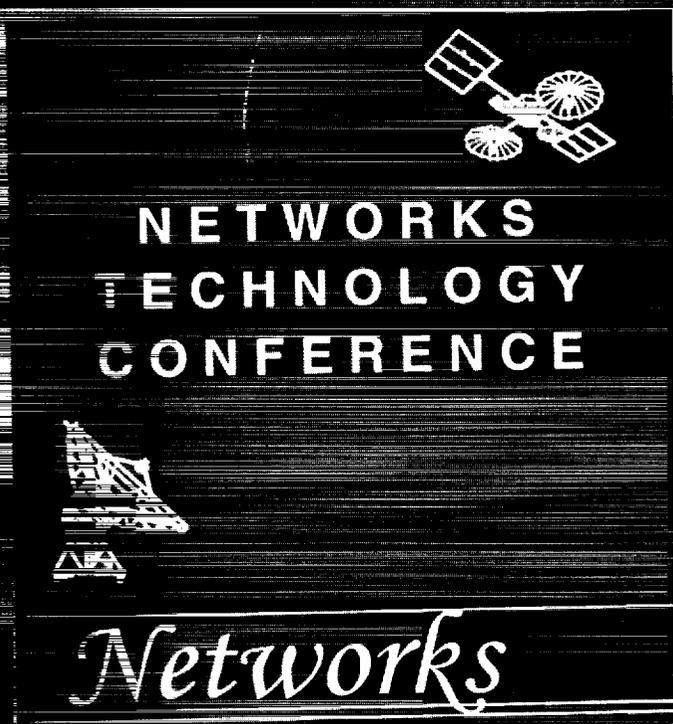


Networks Technology Conference



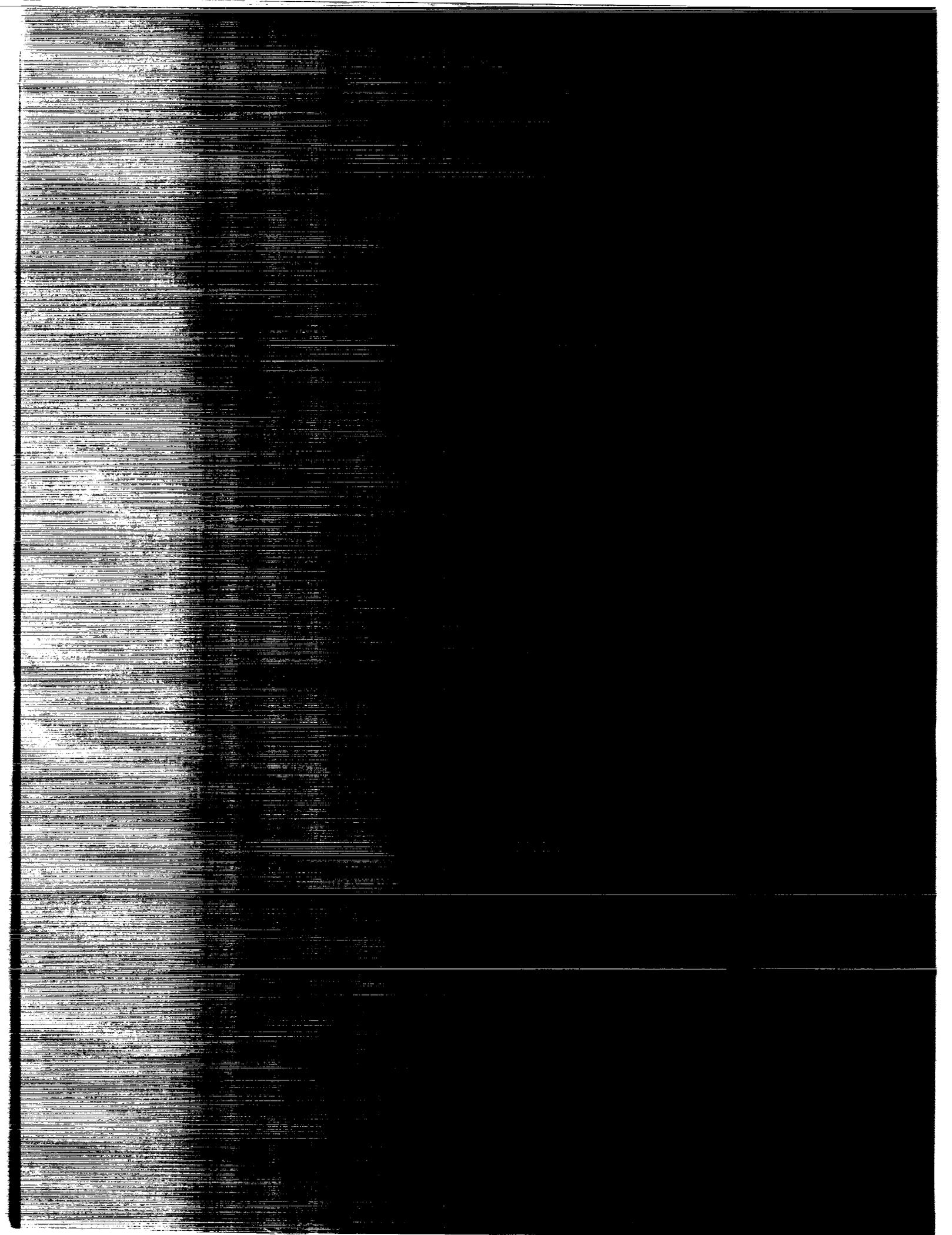
*Proceedings of a conference held at
Goddard Space Flight Center
Greenbelt, Maryland
November 9, 1993*

(NASA-CP-3241) NETWORKS TECHNOLOGY
CONFERENCE (NASA) 364 p

N94-21323
--THRU--
N94-21351
Unclas

H1/62 0197157





NASA Conference Publication 3241

Networks Technology Conference

*Edited by
Keiji Tasaki
NASA Goddard Space Flight Center
Greenbelt, Maryland*

Proceedings of a conference sponsored by the
NASA Goddard Space Flight Center
and held at the Goddard Space Flight Center
Greenbelt, Maryland
November 9, 1993



National Aeronautics
and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

1993

1. The first part of the document discusses the importance of maintaining accurate records of all transactions.

2. It also emphasizes the need for regular audits to ensure the integrity of the data.

3. Furthermore, the document highlights the role of technology in streamlining the accounting process.

4. Finally, it concludes by stating that a strong foundation in accounting principles is essential for any business.

5. In addition, the document provides a detailed overview of the various accounting methods used in the industry.

6. This includes a discussion on the differences between accrual and cash accounting.

7. The document also covers the importance of proper documentation and record-keeping.

8. Overall, the document serves as a comprehensive guide for anyone looking to improve their accounting practices.

Introduction

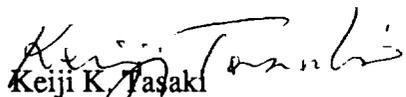
The Networks Technology Conference was conceived as a means to allow all those people involved in the Space, Ground, and Deep Space Network activities to be exposed to the scope of on-going technical work. Additionally, the conference presents an opportunity to showcase the technical accomplishments of personnel of the GSFC Networks Division and their support contractors.

The technical papers included in these proceedings are the result of an intensive effort during the past six months to solicit, evaluate and to select the most interesting and current topics for presentation and publication. A total of 29 papers are included in these proceedings. Regrettably, only 12 of the 29 could be selected for presentation due to time and space limitations. We anticipate that future conferences will be expanded to include a larger participation.

These proceedings are organized to align with the principal technical areas and functions of interest to the Networks Division. These areas are:

- Project Management
- Network Operations
- Network Control, Scheduling, and Monitoring
- Modeling and Simulation
- Telecommunications Engineering

On behalf of the Networks Technology Conference Program Committee, I want to express my appreciation for your participation and interest in the 1993 Networks Technology Conference.


Keiji K. Takaki
Chairman, Program Committee
Networks Technology Conference

Conference Agenda

- 7:30 AM Registration, Coffee, Donuts
- 8:15 AM *Introductory Remarks*
Daniel Spintman, NASA GSFC
- 8:30 AM Session 1 (Papers 1, 2, 3)
- Project Management**
- The NCC Project – A Quality Management Perspective*
Raymond H. Lee
Computer Sciences Corporation
- Network Operations**
- Enhanced Network Operations Using the X-Windows Systems*
Irving Linares
NASA Goddard Space Flight Center
Network Control Systems Branch
- The Automated Ground Network System (AGNS) – An Object-Oriented Design Architecture*
Miles Smith
NASA Goddard Space Flight Center
Telecommunication Systems Branch
- Peter Miltich
AlliedSignal Technical Services Corporation
- 10:00 AM Break
- 10:30 AM Session 2 (Papers 4, 5, 6)
- Network Control, Scheduling and Monitoring**
- Space Network Scheduling Benchmark: A Proof of Concept for Technology Transfer*
Karen L. Moe
NASA Goddard Space Flight Center
Software and Automation Systems Branch
- B. J. Hayden
NASA Goddard Space Flight Center
Network Operations Branch
- Nadine Happell
Stanford Telecom
- Cathy Barclay
AlliedSignal Technical Services Corporation
- Using Manufacturing Message Specification for Monitor and Control at Venus*
W. Randy Heuser
Richard L. Chen
Michael H. Stockett
Jet Propulsion Laboratory
Monitor and Control Technology Group
- Predicting Scheduling Success*
Fredric Messing
Computer Sciences Corporation
- 12 Noon Lunch
- 1:00 PM Keynote Address
- Technology Emphasis in NASA Space and Ground Networks*
Charles T. Force
Associate Administrator
Office of Space Communications, NASA HQ
- 1:30 PM Session 3 (Papers 7, 8, 9)
- Modeling and Simulation**
- The Composite Analytical and Simulation Package for RFI (CASPR) on a Closed Channel*
Jeff Freedman
Ted Berman
Stanford Telecom
- An In-Depth Analysis and Modeling of the Shuttle to MILA S-Band Link*
Armen Caroghtanian
Fernando A. Pellerano
NASA Goddard Space Flight Center
Telecommunication Systems Branch
- Dale D. Shama
Loral Aerosys
- The Automated Conflict Resolution System (ACRS)*
Ted Kaplan
Andrew Mushiner
David Wampler
Stanford Telecom
- 3:00 PM Break
- 3:30 PM Session 4 (Papers 10,11,12)
- Telecommunications Engineering**
- A Packet Switched Communications System for GRO*
Shabu Husain
Wen Hsing Yang
Rani Vadlamudi
AlliedSignal Technical Services Corporation
- Joseph A. Valenti
NASA Goddard Space Flight Center
Spacecraft Control Programs Branch
- Using the USCCS for Sub-Microsecond Spacecraft Clock Calibration*
Victor J. Sank
Fredrick Herold & Associates, Inc.
- A Low Cost Alternative to High Performance PCM Bit Synchronizer*
Bruce DeShong
AlliedSignal Technical Services Corporation
- 5:00 PM Adjourn Concluding Remarks
Keiji Tasaki
NASA Goddard Space Flight Center

CONTENTS

SESSION 1: PROJECT MANAGEMENT

The NCC Project: A Quality Management Perspective	1
<i>Raymond H. Lee</i>	
Let Your Fingers Do the Walking...The Projects' Most Invaluable Tool	11
<i>Deborah A. Zirk</i>	
Software Metrics—The Key to Quality Software on the NCC Project	27
<i>Patricia J. Burns</i>	
NCCDS Configuration Management Process Improvement	41
<i>Kathy Shay</i>	
Problem Reporting Management System Performance Simulation	53
<i>David S. Van Natta</i>	

SESSION 2: NETWORK OPERATIONS

Enhanced Networks Operations Using the X Window System	61
<i>Irving Linares</i>	
The Automated Ground Network System	73
<i>Miles T. Smith and Peter N. Militch</i>	
Mission Operations Support Area (MOSA) for Ground Network Support	83
<i>Robert D. Woods and Susan A. Moser</i>	
Space Network Scheduling Benchmark: A Proof-of-Concept Process for Technology Transfer	103
<i>Karen Moe, BJ Hayden, Nadine Happell, and Cathy Barclay</i>	
Using Manufacturing Message Specification for Monitor and Control at Venus	121
<i>W. Randy Heuser, Richard L. Chen, and Michael H. Stockett</i>	

SESSION 3: NETWORK CONTROL, SCHEDULING, AND MONITORING

Predicting Scheduling Success	137
<i>Fredric Messing</i>	

CONTENTS (cont.)

Distributed Intelligent Control and Status Networking	151
<i>Andre Fortin and Manoj Patel</i>	
Generating and Evaluating Alternate Schedules	163
<i>Surender D. Reddy</i>	
The Test Analysis Retrieval System (TARS): Meeting the Challenges of the Network's Test Processes	181
<i>Robert L. Stelmaszek and Douglas R. Lumsden</i>	
Analysis of the TDRS Multiple Access System for Possible Use as an Attitude Control System Sensor	195
<i>Bruce Allyn Blevins and Victor J. Sank</i>	

SESSION 4: MODELING AND SIMULATION

The Composite Analytic and Simulation Package or RFI (CASPR) on a Coded Channel	209
<i>Jeff Freedman and Ted Berman</i>	
An In-Depth Analysis and Modelling of the Shuttle to MILA S-Band Telemetry Link	217
<i>Armen Caroglanian, Fernando A. Pellerano and Dale D. Shama</i>	
The Automated Conflict Resolution System	225
<i>Ted Kaplan, Andrew Musliner, and David Wampler</i>	
The Three-Dimensional Event-Driven Graphics Environment (3D-EDGE)	235
<i>Jeffrey Freedman, Roger Hahn, and David M. Schwartz</i>	
Developing a Space Network Interface Simulator: The NTS Approach	243
<i>Gary E. Hendrzak</i>	
Satellite Communication Hardware Emulation System (SCHES)	253
<i>Ted Kaplan</i>	

SESSION 5: TELECOMMUNICATIONS ENGINEERING

A Packet Switched Communications System for GRO	265
<i>Shabu Husain, Wen Hsing Yang, Rani Vadlamudi, and Joseph Valenti</i>	

CONTENTS (cont.)

Using the USCCS for Sub-Microsecond Spacecraft Clock Calibration	275
<i>Victor J. Sank</i>	
A Low-Cost Alternative to High-Performance PCM Bit Synchronizers	289
<i>Bruce DeShong</i>	
Array Simulations Platform (ASP) Predicts NASA Data Link Module (NDLM) Performance	309
<i>Allen David Snook</i>	
Conversion of the Proprietary ROLM Inter-Node Link from Multimode to Singlemode Operation	329
<i>Larry Boucher</i>	
Advanced Coding and Modulation Schemes for TDRSS	349
<i>Linda Harrell, Ted Kaplan, Ted Berman, and Susan Chang</i>	
Network Command Processing System Overview	355
<i>Yonwoo Nam and Lisa D. Murphy</i>	



SESSION 1: PROJECT MANAGEMENT



THE NCC PROJECT : A QUALITY MANAGEMENT PERSPECTIVE

Raymond H. Lee
Computer Sciences Corporation
10110 Aerospace Road
Seabrook, Maryland 20706

(301) 794-1600
(301) 552-3272 (FAX)

Summary

The Network Control Center (NCC) Project introduced the concept of total quality management (TQM) in mid-1990. The CSC project team established a program which focused on continuous process improvement in software development methodology and consistent deliveries of high quality software products for the NCC. The vision of the TQM program was to produce error free software. Specific goals were established to allow continuing assessment of the progress toward meeting the overall quality objectives. The total quality environment, now a part of the NCC Project culture, has become the foundation for continuous process improvement and has resulted in the consistent delivery of quality software products over the last three years.

Background

The Network Control Center (NCC) Project has had a long history of developing and maintaining software to assure that the system stays abreast of the changing needs of the Space Network. The NCC must schedule, control, and monitor the real time activities associated with providing communications support to spacecraft requiring the capabilities of the Tracking and Data Relay Satellite System (TDRSS).

In 1989, a major initiative was underway to modernize the White Sands Complex by establishing a Second TDRS Ground Terminal and upgrading the existing terminal to handle the projected workload for the late 1990s. These changes in the Space Network necessitated major changes in the NCC to assure that control center software would be compatible with the ground terminal environment. This development activity was designated as NCC Block 3 and the project was tasked to develop software that would be delivered in three releases. The first release would accommodate the changes necessitated by the Second TDRS Ground Terminal (STGT). The software product would have to operate with formats and functionality that could support the current ground terminal environment and the STGT. The second release was to be designed to handle the STGT and the White Sands Ground Terminal Upgrade (WSGTU). The final release

had the objective of providing the capability to schedule, control, and monitor a four or more TDRS constellation. The planned completion of the NCC Block 3 software upgrades was scheduled for 1995.

In parallel with the challenge to develop the Block 3 software, the NCC Project had to provide software enhancements to the operational baseline to address known system deficiencies and planned hardware upgrades. These activities were to be accomplished under the NCC Block 1 or maintenance effort. Maintenance deliveries were planned for each fiscal year and all operational baseline changes were integrated into the development baselines.

All development planning actions had to take into consideration the impact of the ongoing maintenance tasks.

Management Realignment

Within the context of this phased software development effort, the concept of total quality management was introduced in mid-1990. During the same time frame Goddard Space Flight Center (GSFC) management aligned the NCC Project software development efforts within the Flight Dynamics Division (FDD) from the traditional Networks Division (ND) in the Mission Operations & Data Systems Directorate. This action was taken because it was determined that the appropriate project management structure had to be in place for the NCC Block 3 software development objectives to be successfully met. This structure was not in place in the Networks Division nor was there sufficient expertise to establish the required organization. Thus, a project organization was established within the FDD to manage the CSC SEAS NCC Project Block 3 software development effort.

In concert with the management changes made on the GSFC staff, CSC assigned a new project manager and deputy project manager to lead the NCC Project. This new team established, within the NCC Project, a total quality program which focused on continuous process improvements in the software development methodology and consistent deliveries of high quality software products for the NCC. The vision of the total quality program was to establish and maintain an environment in which software products could be developed and delivered error free. To accomplish this goal, the project had to emphasize its

understanding of customer needs, appreciate the value of continuous process improvement, and recognize the importance of team building and effective communications.

Several measures of merit were established to allow for continuous assessment of the progress toward meeting the overall quality objective. These measures were categorized as the "FOUR P's": Product; Process; Performance; and Participation. The "Product" criteria was measured in terms of the quality of delivered software and the project's ability to deliver on schedule and within budget. The critical quality measures were related to the error rates during system and acceptance testing. The "Process" criteria was measured in terms of process improvements implemented which contributed to product quality and/or cost avoidance. These accomplishments are documented in success stories and task improvement initiatives. The "Performance" criteria was measured by the monthly award fee evaluations. The performance target was to achieve at least 50% plus evaluations during a given fiscal year. The "Participation" measure was the percentage of project personnel who participated in TQM activities. The target was to achieve and sustain a 75% participation rate. Measurements against these goals are accomplished on a monthly basis and feedback is provided to project personnel through newsletters and all-hands meetings.

The NCC Project TQM environment provided the framework for continuously improving the quality of all NCC products and services to meet established goals and objectives. The TQM environment established a highly participative management methodology that complemented the NCC organizational structure. The focus of the methodology was a long-term commitment to improving the quality of products and services. This methodology used a hierarchy of NCC management-chartered committees to improve the quality of NCC processes and products. These committees accomplished this by identifying, assessing, measuring, analyzing, and documenting processes, issues, and products and recommending improvement action plans to management.

The TQM program was implemented through the NCC Quality Management Board (QMB), designated permanent committees and selected ad hoc committees. The five permanent committees include the Suggestion Analysis and Resolution Group (SARG), the Customer Satisfaction Committee (CSAT), the Worklife and

Communications Committee (WLCC), the Process Improvement Committee (PIC), and the Awards and Recognition Committee (ARC). The SARG was chartered to analyze suggestions for improving products and processes. Following the analysis, a recommendation for implementation would be provided. The CSAT was chartered to establish consistent and meaningful mechanisms for obtaining client feedback and measuring customer satisfaction. The WLCC was chartered to develop, implement, and facilitate an effective NCC communications program. The PIC was chartered to focus on methods to improve technical processes. Many of the ad hoc Process Action Teams (PATs) evolved from PIC initiatives to update or modify existing procedures. The ARC was chartered to develop, implement, and facilitate the NCC recognition program. Their work was the basis for the monthly recognition program at all-hands meetings and the annual NCC DATUM award to the project employee of the year.

The Total Quality Impact

Given the scope of the tasks that the project had to accomplish, the organizational environment (i.e. the FDD GSFC management team), and the newly established TQM environment for the SEAS NCC project team, a look at the results and on-going activity provides an interesting perspective on how a large software development project can succeed.

One of the most important contributors to success was responsiveness. This was demonstrated in the initial transition of the project to the FDD. Close coordination and communication among all parties, intense effort, and a strong desire to succeed paved the way to assure a smooth transition took place. Responding to five new task assignments which mapped to the development, maintenance, system engineering, system testing, and security requirements, the NCC project team replanned all the work in less than one month. To expedite task planning, CSC management and technical personnel worked closely with GSFC managers to familiarize them with the NCC system architecture, software environment, and system functionality. Frequent meetings between the project task managers and their GSFC counterparts ensured that the technical effort mapped correctly to the new task structure.

During the transition, project personnel were preparing to meet two critical milestones. On August 7, 1990, the CDR for Block 3, Release 1 was held

successfully. Related deliveries included the detailed design document, operations concept document, quality assurance plan, system test plan, and configuration management plan - collectively, more than 4000 pages. This event concluded the detailed design phase and clearly demonstrated the readiness for software implementation. The relatively small number of Review Item Disposition (RIDs) (31), none of which affected the overall design direction, testifies to the high quality of the design. The second event was the delivery on September 17 of the FY90 maintenance release to acceptance testing 2 weeks early. The quality of the software was evident from the low number of problem reports (11) identified and corrected during system testing. This release, designated 90.1, completed acceptance test one month earlier than scheduled. Release 90.1 was the first delivery to operations in which the label "error-free" could be applied since there were no errors associated with new or modified functions.

As the project stabilized in the FDD environment, significant milestones continued to be met. The FY91 maintenance release, 91.1, completed system testing as scheduled. The upgrade of the Communications and Control Segment (CCS) VAX to an 8550 processor configuration was system tested and integrated into the operational configuration. The Block 3, Release 1, Build 1 activity completed integration testing on schedule and made significant progress in implementing Build 2 during the spring of 1991. Release 91.1 was delivered to operations during the summer of 1991. This release consisted of operating system upgrades to both the UNISYS and DEC processors. These upgrades were implemented as a risk mitigation action for Release 1 to minimize the amount of change occurring with a planned development release. The lesson learned was that operating system changes in a real time system environment can have unexpected results as was evident during transition to operations. Both timing and performance deficiencies were uncovered which required quick fix actions to maintain system stability. The final delivery for release 91.1 was accomplished in November 1991.

In parallel with the maintenance activity, the Block 3, Release 2 CDR was successfully held in October 1991, while system testing of the Release 1 final build was progressing as planned. As another risk mitigation activity, the system test team performed limited interface testing with STGT while it was at the vendor's site. These tests proved invaluable in uncovering problems early

and assigning responsibility for resolution. This activity contributed to the establishment of the interface incident reporting (IIR) process which is now in place to document and resolve interface incompatibilities within the Space Network. Block 3, Release 1, was delivered to acceptance test in March 1992 on schedule and within budget. The release was over 100,000 Delivered Source Instruction (DSI) and a relatively small number of problem reports were identified during acceptance testing. After a successful ORR on August 27, 1992, the Block 3, Release 1 (Release 92.1) was successfully transitioned to operations on September 25, 1992.

To achieve this successful delivery over a two year period, several significant process improvements were put in place, resulting in a more disciplined software development environment and a significantly higher quality product. During implementation, all changes were subjected to a rigorous design, code, and unit test inspection and certification process. The configuration management team implemented procedures that reduced baseline build errors by over 50 percent. The system test team developed more detailed functional and regression test procedures that reduced the number of errors found by the acceptance test team by over 25 percent of that projected for a release of this size. Overall, the software engineering process improvements implemented during the Release 1 development cycle have resulted in a capability for the NCC to be compatible with all Space Network elements and to operate more reliably in the STGT era.

Block 3, Releases 2 and 3 are still under development and progressing as planned. Release 93.1, the FY93 maintenance release, was successfully delivered to operations on August 25, 1993 with the highest quality measure for any NCC delivered product.

The Transition Back To Networks Division

After two and a half years, the software development effort was transitioned back to the Networks Division. The lessons learned from the Flight Dynamics software management experience were an integral part of the successful transition back to the Networks Division. A comparable GSFC management team was established to work with the Computer Sciences Corporation (CSC) NCC Project management team.

In order to accomplish a successful transition, a plan was developed in which a period of 9 months was allocated for transition activities. The first step was to establish a comparable GSFC technical management team in the Networks Division. This team was carefully selected and in April 1992 they began attending the weekly task meetings. This provided a mechanism for the new team to become familiar with the technical issues, the development methodology, and the technical management roles and responsibilities. During the first 6 months of the transition period, the FDD team was in the decision making role for the project. During the last 3 months, the ND team took the lead while the FDD team assumed the role of adviser and observer. At the beginning of 1993, the transition was completed without impact on any critical NCC milestones.

Key Lessons Learned

The key lessons learned were in four areas: (1) commitment at all levels in the project - both CSC and GSFC; (2) requirements control and effective change management vital; (3) product focus as well as process focus is key to success; and (4) consistent measurement of critical success factors.

The management commitment to the NCC Project was visible and sincere on the part of GSFC and CSC. In the spring of 1990, CSC not only selected a new management team for the project but also established periodic meetings between senior MO&DSD personnel and CSC corporate personnel. These meetings were held quarterly and provided the forum for discussion of issues which might impact the project's objectives. The participants at these meetings included the Director and/or Deputy Director of MO&DSD, the CSC Systems Group President, the Systems Sciences Division President, the SEAS Program Manager, and the SEAS Deputy Program manager. This level of commitment assured that the appropriate level of attention and resources would be applied should an issue arise which could potentially impact the project. The periodic meetings continued through the transition of the project back to the Networks Division and will be convened in the future as required.

Requirements control and change management were vital to the success of the project. The management decision to separate the software development activities from the requirements definition functions was critical to requirements control. The GSFC FDD management team had the responsibility to focus on the software development progress and control the growth of the system. As new or changed requirements were identified by the ND NCC Project Office, an impact assessment was performed and a conscious decision was made to consider adjusting established milestones and resources to accommodate new requirements. In addition to the external pressures to change requirements, there were internal proposed changes that were identified during the design and development phases of the implementation. To control these changes, processes were put in place for both technical and management reviews of all proposed changes through the Technical Review Board (TRB) and Configuration Review Board (CRB). These internal project reviews are an integral part of the change management process and provided a more rigorous approach to managing the scope and size of the software effort.

With the establishment of the Total Quality Management Program, there was significant emphasis on continuous process improvement. However, in parallel with that emphasis, there was need to keep the focus on product improvement as well. This was accomplished by establishing a clear cut vision which focused on the product and challenged project personnel to develop and deliver error-free software. By having this focus, all impacts of process improvement activities could be measured either directly or indirectly with the quality of the delivered release. The result of the product focus can be seen in Figure 1 which shows the error rate of the major releases over the last four years. The results of the process improvements are readily seen in the trends showing a steady decrease in the error rate during system and acceptance testing.

The final area where lessons were learned was in the establishment of a measurement program and sustaining it to provide visibility into the project's progress toward meeting targeted objectives. The "Product" measure as stated above was the error rates experienced during the test phases. This is illustrated in Figure 1. The impact of the quality initiatives is clearly seen in the error rate in releases 92.1 and 93.1 which were .24 and .19 errors/KDSI respectively. The "Performance" measure is shown in Figure 2 where the goal

of 50% plus evaluations is on target through fiscal year 1993. The "Process" measure of success stories is demonstrated by the fact that during FY1993, there have been 18 success stories written by project members with a total cost avoidance value of \$ 387,000 . The final measure was "Participation". During the current fiscal year, participation has exceeded 85% of project personnel. These measures have been used for management assessment of the TQM program, but these initiatives have had a much deeper impact on the project. The total quality environment is now a part of the NCC Project culture. It remains the foundation for continuous process improvement on the project and is the basis for the consistent quality of all NCC software products.

FIGURE 1
NCC RELEASE ERROR RATES

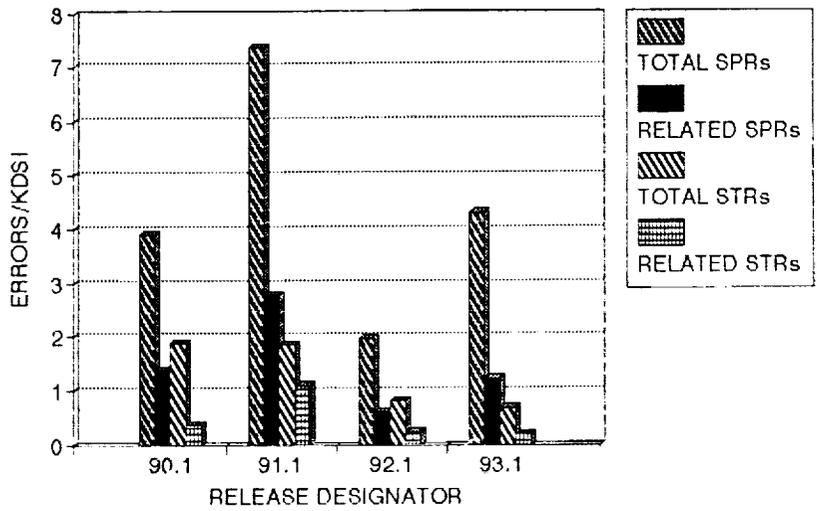
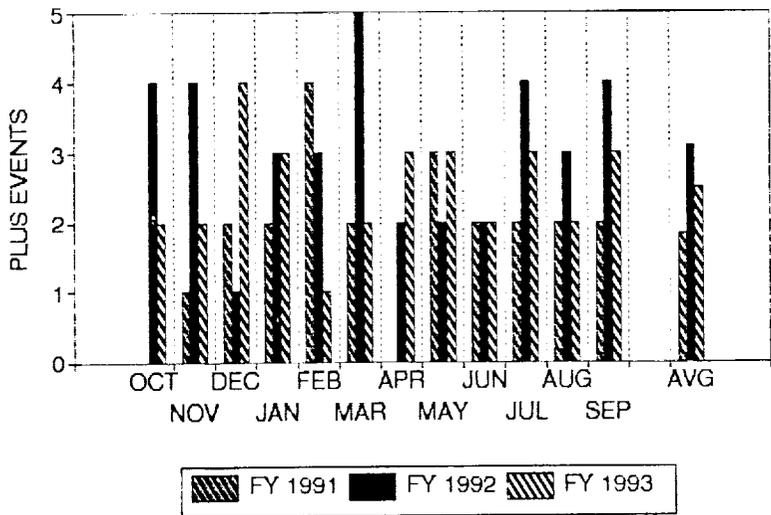


FIGURE 2
NCC PROJECT PERFORMANCE



LET YOUR FINGERS DO THE WALKING
THE PROJECTS MOST INVALUABLE TOOL

Deborah A. Zirk
Computer Sciences Corporation
GreenTec I Building
10000 K Aerospace Road
Seabrook, Maryland 20706
(301)794-1383 (Office)
(301) 552-3272 (FAX)

1. INTRODUCTION

The barrage of information pertaining to software being developed on a project can be overwhelming. Current status information as well as the statistics and history of software releases should be "at the fingertips" of project management and key technical personnel. This paper discusses the development, configuration, capabilities, and operation of a relational database, the System Engineering Database (SEDB) which was designed to assist management in monitoring of the tasks performed by the Network Control Center (NCC) Project. This database has proved to be an invaluable project tool and is utilized daily to support all project personnel.

1.0 DEVELOPMENT METHODOLOGY

The development of the SEDB utilizes a spiral methodology, whereby capabilities are prototyped, implemented and testing concurrently. Each capability provided by the SEDB is "prototyped a little, implemented a little, tested a little, prototyped a little more, implemented a little more, and tested a little more". Throughout development the primary users of each capability are significantly involved by utilizing the prototype and providing comments. This reiterative methodology prevents the full implementation of a capability that is not useful or appropriate for the users. The use of such a methodology has been shown to be significantly efficient and effective for all types of software development.

The initial SEDB development was to provide a tracking mechanism for impact assessments. Its original purpose was merely to provide a database in which

these impact assessments would be logged. As this capability was being prototyped, Quality Assurance (QA) was in need of a better problem reporting database. The impact assessment capability was therefore modified and prototyped again to include the problem report capability. This initial development determined the methodology for all future database efforts. Project personnel determine a need and work closely with the database developers in ensuring its correct implementation.

2.0 CONFIGURATION

The SEDB is capable of supporting a project in a standalone mode or in a LAN configuration. A standalone configuration only allows one user at a time access to the database. The requirements for a standalone configuration are as follows:

- o 486DX33 PC
- o 8 MB RAM
- o 200 MD HD
- o Laser Printer
- o DOS 5
- o RBase Version 4.0

To allow multiple users simultaneous access to the database, a LAN configuration is required. The NCCDS Project utilizes the following LAN configuration:

Workstation:	486DX33
	8 MB RAM
	200 MD HD
	EtherNet Card
Fileserver:	486DX 33
	8 MB RAM
	500 MB HD
	EtherNet Card
	Laser Printer

The software requirements are as follows:

DOS 5
Novel Netware

RBase Version 4.0

The LAN components (10 base - twisted pair) are as follows:

HUB
Patch Panel
Telco Block
Level 4 Wiring

The SEDB can be run with a copy of RBase Runtime (this is available at no cost from the author), thereby eliminating the need to purchase RBase 4.0. RBase Runtime, however, does not allow software modifications on the PEDB.

3.0 SYSTEM ENGINEERING DATABASE SEGMENTS

The SEDB is comprised of four major segments: the Project Engineering Database (PEDB), the Configuration Management Database (CMDB), the Requirements Database (RQDB), and the Hardware Resources Database (HWDB). This section specifically describes each of these segments and the benefits each segment provides to the project.

3.1 Project Engineering Database (PEDB)

The PEDB segment is the most heavily utilized segment of the database. It manages all project problem reporting data such as System Trouble Reports (STRs), System Problem Reports (SPRs), and Integration System Problem Reports (ISPRs). In addition, the PEDB also is used to manage all other potential project impacts, such as NCC Requirements Inputs (NRIs), Minispecifications, and future impacts. The PEDB also provides the user with the capability to submit/review SEDB database problem reports. The PEDB allows project management to retrieve this data for:

- o History information (e.g., How long did it take Development personnel to turnaround fixesto problems found during System Testing for a particular release two years ago? What was the Development turnaround time for a release six months ago?)
- o Current status of a particular problem/impact entity/release (how many ISPRs have been closed currently in Integration Testing?)
- o Planning purposes (e.g., how many DSI does a future release contain if we include the resolution to these STRs?)

The PEDB offers four significant benefits to a project. One major benefit is that it allows searching of problem reports and impact entities by any data field. This allows the database user to find what he/she needs using any information available. For example, a problem report can be searched by number, a text string, system functionality, responsible organization, problem type, priority, status, and/or system configuration.

Secondly, the PEDB automates the preparation of reports which can be displayed and/or printed. These reports can be created by the database user utilizing the search capability to generate the information desired or by utilizing the standard reports currently existing on the PEDB. One example of a standard report that is of significant importance to a project is the weekly release status report (see Figure 1) which is automatically generated using the PEDB. This report provides detailed information about the status of all problems for a particular release.

A third major benefit of the PEDB is that it provides current, consistent statistical data. Since all the project data is kept on one database, all project personnel have access to the same data and the assurance that the data is the most current.

Finally, the PEDB provides a historical archive for analysis. All data for previous releases can be accessed through the PEDB. This historical data can be utilized to provide release history information and defect causal analysis data (e.g., the number of problems written against each subsystem, the average length of time required to resolve each problem, the number of problem reports by problem type, etc.).

3.2 Configuration Management Database (CMDB)

The prime responsibility of the CMDB segment is to track software through the development process, that is, to manage CM information. The CMDB segment was created to replace manual CM logs. CM personnel enter data into the CMDB directly from internal software delivery forms received from Development personnel. This allows the CMDB to generate and identify delivery contents for a specific build and/or release for a specific time period (see Figures 2 and 3). In addition, the CMDB allows the monitoring of CM level status for all units,

Release Leader: 2. Johnson
 BILLED 0 SPR STATUS REPORT For RELEASE:3 AS OF:09-16-93 PAGE:23.
 Assigned
 To
 930621-0242 3 A. MORGAN
 SPS DEV

SV DOC DAT SYS CM RUIID Dev
 Chg Chg Chg Chg Chg Chg Req Code Facility
 Status Chg Chg Chg Chg Chg Chg
 CLOSED MO MO MO MO MO MO
 08-24-93

Description
 ACK RECEIVED FLAGS THAT ARE SET TO "YES", FOR SGT AND USC, BY AIS TRANSMISSION
 ARE RESET TO "NO" AFTER AN SPS CONDSTART. NOTE:WHEN ACK RECEIVED FLAGS ARE SET
 TO "YES" BY TRANSMISSIONS TO EACH GT SEPARATELY THE FLAG REMAINS SET TO "YES"
 AFTER CONDSTART.

STATUS
 DATE WRITTEN 06-18-93
 DATE RECEIVED 06-21-93
 ASGM TO SPS DEV 06-21-93
 OA 06-25-93
 TO RL LEVEL 1 06-25-93
 TO CM LEVEL 1 06-29-93
 TO IT 06-29-93
 IT CLOSED 07-07-93
 TO RL LEVEL 2 07-07-93
 TO CM LEVEL 2 07-07-93
 TO ST LEVEL 2 07-08-93
 ST CLOSED 08-24-93

SPR 930623-0243 4 DESANILS/MORGAN
 CCS/SPS DEV
 CHAPTERS 1-5 OF THE C.O.U.C. DO NOT REFLECT THE CHANGES MADE IN RTRD. FOR
 EXAMPLE, PAGES 3-6 AND 3-5 HAVE NOT BEEN RECLAIMED (ADT REFERENCES NOT DELETED).
 MOR HAS PAGE 3-67 (GETABLE DISPLAY LIST NOT CHANGED).

CLOSED MO MO MO MO YES
 09-10-93

STATUS
 DATE WRITTEN 06-23-93
 DATE RECEIVED 06-23-93
 ASGM TO DEVEL 06-23-93
 OA 07-23-93
 TO RL LEVEL 1 07-23-93
 TO CM LEVEL 1 07-23-93
 TO IT 07-27-93
 IT CLOSED 08-11-93
 TO RL LEVEL 2 08-11-93
 TO CM LEVEL 2 08-11-93
 TO ST LEVEL 2 08-11-93
 ST CLOSED 09-10-93

SPR 930625-0244 3 D. DESANILS
 CCS DEV
 AFTER DOWNING THREE SITES FOR WHICH UPD HAD BEEN SELECTED, MM RECEIVED AN
 INFORMATION ALERT, "MM-PERF DATA TRANSMIT PROBLEM CONTINUING TO 09/03 SELECTED
 SITES" WHERE 09 IS THE NUMBER OF THE SELECTED SITES WITH TRANSMISSION PROBLEMS.
 THE 09 IN THE INFORMATION ALERT SHOULD READ 03.

CLOSED MO MO MO MO
 08-27-93

STATUS
 DATE WRITTEN 06-24-93
 DATE RECEIVED 06-25-93
 ASGM TO CCS DEV 06-25-93
 OA 07-13-93
 TO RL LEVEL 1 07-13-93
 TO CM LEVEL 1 07-13-93
 TO IT 07-13-93
 IT CLOSED 07-15-93
 TO RL LEVEL 2 07-15-93
 TO CM LEVEL 2 07-15-93
 TO ST LEVEL 2 07-15-93
 ST CLOSED 08-27-93

Figure 1. Example of a weekly release status report generated by the Project Engineering Database (PEDB).

Date : 05/28/93

List of XXX units delivered for RELEASE 93.1

Unit Name	Type	Action
APPGBL.MMS	M	C
BELLTA	S	d
BELLTARA	C	A
CALTSYN	S	C
CCCFERP	S	C
CCCFERP1	S	C
CCCFERP2	S	C
CCCINIT	S	C
CCCISBU	S	C
CCCISCT	S	C
CCCISRP	S	C
CCCISRP1	S	C
CCCISRP2	S	C
CCCISUB	S	C
CCCISYN	S	C
CCCISYN1	S	C
CCCLSPS2	S	A
CCCNORQ	S	C
CCCPTUD	S	C
CCCRAPP	S	C
CCCRAPS	S	C
CCCRASD	S	C
CCCRASW	S	A
CCCRPRE	S	A
CMCCDY	S	C
CMCDYN	S	C
CMCICS	S	C
CMCNCS	S	C
CMCRAP	S	C
CMCSDP	S	A

End of data for XXX (where XXX = Segment)

Unit Type = S(ource), T(emplate - SPS), P(roc - SPS),
Q(LP Report - SPS), H(Schema/Subschema - SPS),
R(unstream - SPS or ITS, (Mapstream - SPS),
I(nclude File - CCS, NFE, ITS), D(ata File - CCS, ITS),
C(ommand Procedure - CCS), M(MS Description File - CCS,
or O(ther)

Action = A(dd), C(hange), or D(elete)

Page : 1

Figure 2. Example of a Configuration Management Database (CMDB) report listing units delivered for a release.

S D F S T A T U S R E P O R T

This report lists fixes installed in the SDF for RELEASE 93.1 since the start of System Test.

Report Date : 5/28/93

CCS - Current Diskpack : 93_1

ITS - Current ID : 93.1

SPS - Current Onlines : NCC*NCC.ONLINE
NCC*NCC.SPS

NFE - Current ID : 93.1

REQUIREMENT	DATE DELIVERED	SEGMENT	LEVEL
SCR 71	2/9/93	CCS	N/A
SPR 92-12-01-303	2/9/93	CCS	N/A
SPR 92-12-09-312	2/3/93	CCS	2
SPR 92-12-21-331	2/11/93	CCS	2
SPR 93-01-22-013	4/5/93	CCS	2
SPR 93-01-25-022	2/11/93	CCS	2
SPR 93-02-02-040	3/28/93	CCS	2
SPR 93-02-02-041	3/28/93	CCS	2
SPR 93-03-16-114	4/9/93	CCS	2
SPR 93-03-17-119	4/1/93	CCS	2
SPR 93-04-13-164	4/13/93	CCS	2
SPR 93-04-26-0186	4/28/93	CCS	N/A
SPR23	5/31/93	CCS	2
MINISPEC IT032	1/15/93	ITS	2
SPR 92-12-10-318	2/25/93	ITS	2
SPR 92-12-31-336	1/15/93	ITS	2
SPR 93-01-18-010	1/29/93	ITS	2
SPR 93-02-19-069	4/7/93	ITS	2
SPR 93-03-15-110	3/25/93	ITS	2
93.0QF1 MERGER	4/23/93	SPS	2
Runstreams	12/14/92	SPS	N/A

* - New item within the last week

If you have any questions regarding this report, please contact <processor's name> in <processor's location> or at <phone number>.

Figure 3. Example of a Configuration Management Database (CMDB) report listing the status of problem resolutions installed in the Software Development Facility (SDF).

i.e., it verifies that the appropriate units are propagated to specific testing levels.

The CMDB provides three major benefits to the project. One significant contribution is that the CMDB provides QA with a Software Engineering Notebook (SEN) verification tool. QA can utilize the CMDB to ensure that Development SENs are accurate by comparing such information as delivery dates, frequency of unit deliveries, and type of unit changes.

A second contribution of the CMDB is that it provides CM with a much needed internal organizational tool. It eliminates all costs associated with maintaining manual logs. In addition, the CMDB maintains easily accessible CM historical information for analysis purposes.

Finally, the CMDB provides an online reporting capability for CM. Specifically, CM personnel can automatically generate reports for (1) units delivered for a specific build/release, (2) deliveries processed at any point in time, and (3) elapsed time between when CM personnel received a delivery and when it was processed.

3.3 Requirements Database (RQDB)

The RQDB segment manages the text of the NCCDS Detailed Requirements Document (530-DRD-NCCDS). Specifically, the RQDB provides on-line accessibility to the requirements to all project personnel. The RQDB offers browse and search capabilities, thereby allowing project personnel to find on-line a requirement(s) text by paragraph number and/or a text search. The RQDB allows only approved personnel to update the database with approved Document Change Notices (DCNs) and Revisions. All project personnel, however, have the capability to enter comments on each requirement. These comments are saved as attachments and are normally used to provide information on software versus operational requirements, and/or impact assessments performed on a specific requirement.

The RQDB also manages the relationship of the system software and test cases to requirements. Specifically, all system Computer Software Configuration Items (CSCIs), units, modules and executables are mapped to individual requirements. The lines of code associated with each software unit are also maintained in the

RQDB. The capability to perform an online search for units is available to users. As a system security measure, users are allowed to download this software data but are prevented from uploading data. In support of system testing, the RQDB provides a mechanism to map a test case to each requirement, thereby allowing the automatic generation of test and implementation matrices (see Figure 4).

The RQDB provides four significant benefits to the project. The first benefit is the result of the requirements being automatically updated by downloading the requirements document file(s) into the database. This capability ensures complete accuracy of the requirements.

Second, the RQDB allows project personnel to perform string searches to identify associated requirements to the specific requirement being examined. This capability allows individuals to quickly find all requirement references to a specific item.

A third major benefit is the capability to automatically create implementation and test matrix reports. This feature provides reports that list each requirement, the associated CSCI, the test case and type, and the implementation date. This information is updatable for each release and allows efficient generation of reports for various requirements and design reviews.

Finally, the RQDB provides the capability to access online Delivered Source Instruction (DSI) counts. Each requirement is mapped to its associated DSI. This capability allows personnel to quickly determine the impact of implementing or changing a requirement.

3.4 Hardware Resource Monitoring Database (HWDB)

The most recently developed segment of the SEDB is the Hardware Resource Monitoring Database (HWDB). This segment was developed to provide for entry of data related to hardware resource usage time. As the NCC utilizes two development facilities [the Software Development Facility (SDF) and the Development Test and Training Facility (DT&T)], each with several types of equipment, the HWDB provides comprehensive resource utilization data for a specific time period (e.g., weekly, monthly). This data provides the time

NCCDS REQUIREMENTS IMPLEMENTATION REPORT

Requirement #	Para. #	Requirement	Verif.	Allocation	Impl. Date
SECTION 5. SERVICE PLANNING					
5.1 INTRODUCTION					
5.2 PREREQUEST ACTIVITIES					
5.2.1 GENERAL					
05020104-0100	5.2.1.4	The NCCDS shall have the capability to store initial load data, configuration codes, and prototype events for 100 spacecraft in the NCC data base. --NOTE-- In the Baseline NCCDS, the effective capacity is 50 spacecraft.	T	14022000 14024000 14028000 54061000	10/15/94 10/15/94
05020105-0100	5.2.1.5	To facilitate the management of the SI scheduling data base, the NCC DB shall be provided with the following capabilities:			
05020105-0100a	5.2.1.5	Enter a new Spacecraft Identification Code (SIC) and related spacecraft data into the data base.	T	14023000 14026300	10/15/94 10/15/94
05020105-0100b	5.2.1.5	For each SIC, the related spacecraft data includes a list of valid support identification codes (SUPIDENS).	T	14023000	10/15/94
05020105-0100c	5.2.1.5	Delete a SIC and related spacecraft data from the data base.	T	14023000 14026300	10/15/94 10/15/94
05020105-0100d	5.2.1.5	Modify spacecraft data within the data base.	T	14026400 14026500	10/15/94 10/15/94
VERIF. METHOD CODES: A - Analysis D - Demonstration I - Inspection T - Test					
ALLOCATION: * In all cases, the most significant four digits of allocation are HWCI or CSCI number. Additional characters (if any) specify a software design structure. In most cases, such a software design structure is a software component (SC).					

Figure 4. Example of the NCCDS implementation report generated by the Requirements Database (RQDB).

requested by the user, time allocated, and time actually used. In addition, the HWDB maintains lost time data and provides the capability to enter lost time reason codes by day and by hardware resource type. Based on the above information, the HWDB provides management a tool to review or automatically generate reports via several options including, but not limited to: organization, facility, time period, requestor, release, and lost time summary (see Figure 5).

The primary benefit provided by the HWDB is that the maintenance of hardware usage data provides a history of resource utilization. This information is useful for determining how many resource hours are being lost and the reasons for the lost time. In addition, resource utilization metrics are useful for projecting future resource needs for similar releases.

4.0 OPERATIONAL USAGE OF SEDB

The SEDB may be utilized by all project personnel. Figure 6 illustrates an overview of SEDB usage. Specifically, the following provides the normal utilization of the SEDB for each project group:

- o QA - enter/update SPR and STR data
 - compare delivery contents generated from the CMDB to Development SENS (validation tool)
- o CM - enter/update software baseline data
 - verify appropriate units propagated to specific testing levels
 - report CM data
- o Test - enter test case data for requirements
 - generate test status information
- o Development - review SPR and STR status
 - review impact assessment information
- o Maintenance - review SPR and STR status
 - review impact assessment information

RELEASE: <ALL>
 ORGANIZATION: <ALL>

NCC/DT&T H/W RESOURCE
 FOR WEEK BEGINNING
 08/16/93
 SCHEDULE(S) IN HOURS

As of 09/02/93

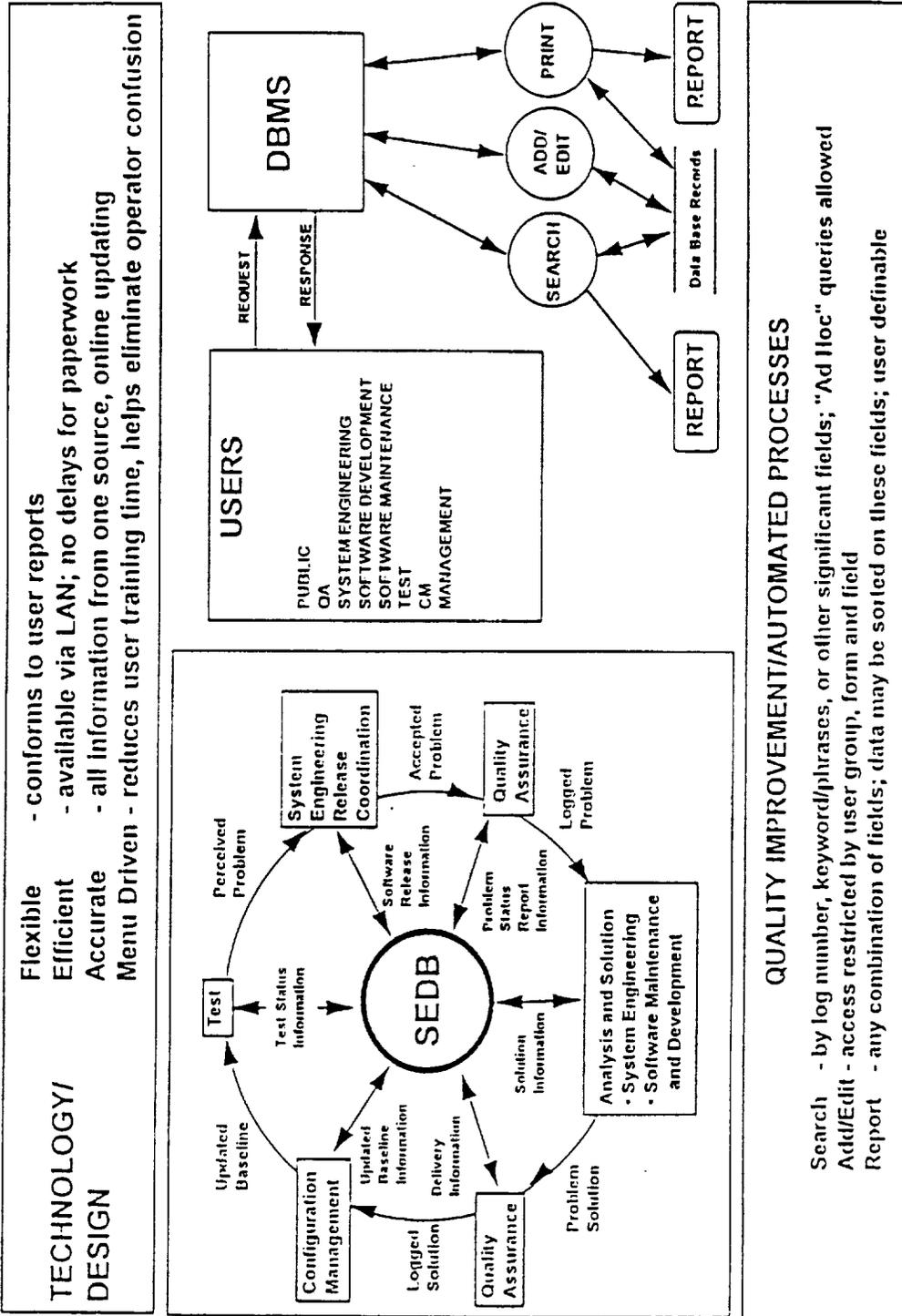
DAY	RAP3		CCS		ITS		IT 15		SPS		NTS		OCR		ENCC							
	RT	AT	RT	AT	RT	AT	RT	AT	RT	AT	RT	AT	RT	AT	RT	AT						
Mon	24	24	20	18	16	14	12	14	12	4	14	12	9	14	12	12	0	0	0	0		
Tue	20	20	32	26	28	14	14	14	12	6	6	26	20	22	20	22	0	0	0	0		
Wed	48	48	38	32	32	14	14	14	12	6	6	26	20	20	20	20	0	0	0	0		
Thu	30	24	26	14	14	14	14	14	12	0	0	26	14	14	14	14	0	0	0	0		
Fri	30	24	26	14	14	14	14	14	12	0	0	26	14	14	14	14	0	0	0	0		
Sat	24	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Sun	24	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
WK	200	188	142	104	104	70	68	70	62	24	16	118	80	79	118	80	82	0	0	0	0	
MCh	392	380	270	214	214	198	178	180	152	96	88	246	190	189	264	208	210	65	65	65	0	0

REQUESTOR	ORGANIZATION	RELEASE
RUPCH	Acceptance Test	N/A
LISA	Development Test	93.1.1
FELL	Maintenance	N/A
VAN METER	Operations	N/A
OPS	System Test	93.1.1
GAIL	System Test	R3B0
GAIL	System Test	N/A
WOLFF	System Test	N/A

RT = Requested Time
 TR = Time Received
 AT = Actual Time Utilized

Figure 5. Example of a Hardware Resource Monitoring Database (HWDB) report for a one week period.

Systems Engineering Data Base



QUALITY IMPROVEMENT/AUTOMATED PROCESSES

- Search - by log number, keyword/phrases, or other significant fields; "Ad Hoc" queries allowed
- Add/Edit - access restricted by user group, form and field
- Report - any combination of fields; data may be sorted on these fields; user definable

Figure 6. Overview of Systems Engineering Database (SEDB) usage.

- o System Engineering - enter/update impact assessment information
 - enter/update defect causal analysis data
 - generate DSI counts for requirement impact assessments
 - create software release status reports
 - enter/update resource utilization information
- o Management - review software release status
 - review SPR/STR status
 - review resource utilization metrics

Access to the SEDB for any add/edit capability is restricted by user group, report form, and data field. This ensures that all information contained within the database is protected from unauthorized access.

The SEDB allows all users to search data by log number, keyword/phrases or other significant field. This provides a mechanism for quick access to specific information.

In addition, the SEDB allows reports to be generated based on any combination of fields, that is, data may be sorted on these fields. Users can define what information is desired for a specific report.

The operational usage of the SEDB provides significant advantages to the project. It is flexible - it allows reports to be quickly created for specific users. The SEDB is efficient - it is immediately accessible via the LAN, therefore there is no need for paperwork delay. Since all information is from one source with an online updating capability, the SEDB is also accurate. All project personnel have access to the same information. Finally, since the SEDB is menu driven, user training time is minimal and operator confusion is virtually eliminated.

5.0 FUTURE ACTIVITIES

It is anticipated that the SEDB will be utilized on other projects; for example, it has been successfully transported and utilized on the X-Band Synthetic Aperture Radar (XSAR) project. Although the future plans for the SEDB depend upon funding, the following represent some of the potential capabilities that are being examined for possible implementation:

- o Display Graphics and Graphical Outputs
- o Personnel Resource Management Database
- o Online Interface Control Documents (ICDs) and Data Format Control Documents)
- o Cost Estimation

6.0 SUMMARY

In summary, the SEDB is an excellent tool that provides access to standardized and consistent data. It facilitates management by allowing the monitoring all tasks within their purview and provides a method to produce task required products in a timely, accurate and consistent formats. It can be utilized by the entire project and provides a basis for continuous process improvement.



Software Metrics - The Key to Quality Software on the NCC Project

Patricia J. Burns
Computer Sciences Corporation
10110 Aerospace Road Seabrook, Maryland
301.794.1640 (o) 301.552.3272 (fax)

SUMMARY

Network Control Center (NCC) Project metrics are captured during the implementation and testing phases of the NCCDS software development lifecycle. The metrics data collection and reporting function has interfaces with all elements of the NCC project. Close collaboration with all project elements has resulted in the development of a defined and repeatable set of metrics processes. The resulting data are used to plan and monitor release activities on a weekly basis. The use of graphical outputs facilitates the interpretation of progress and status.

The successful application of metrics throughout the NCC project has been instrumental in the delivery of quality software. The use of metrics on the NCC Project supports the needs of the technical and managerial staff. This paper describes the project, the functions supported by metrics, the data that are collected and reported, how the data are used, and the improvements in the quality of deliverable software since the metrics processes and products have been in use.

NCC PROJECT OVERVIEW

The NCC is an element of the National Aeronautics and Space Administration (NASA) Spaceflight Tracking and Data Network (STDN). The STDN is a worldwide complex designed to provide tracking and data acquisition support to manned and unmanned spacecraft in low-earth orbit. It is composed of the Space Network (SN) and the Ground Network (GN). The STDN has evolved into a network that currently uses the Tracking and Data Relay Satellite System (TDRSS) as the primary source of support for orbiting spacecraft. The NCC Project Office, Goddard Spaceflight Center (GSFC) Code 530, is responsible for the support and maintenance of the current NCC Data System (NCCDS). The NCCDS performs network scheduling, acquisition and tracking support, data quality assurance, performance monitoring, and overall coordination of the STDN.

The NCCDS is maintained by the NCC Project within the Computer Sciences Corporation (CSC) System Sciences Division (SSD) Networks Technology Group (NTG). The maintenance and enhancement of the NCCDS is performed as part of the System

Engineering and Analysis Support (SEAS) contract. These responsibilities have been distributed among several tasks: Software Development, Software Maintenance, Integration Test, System Test, System Engineering, Metrics, Configuration Management, and Product Assurance.

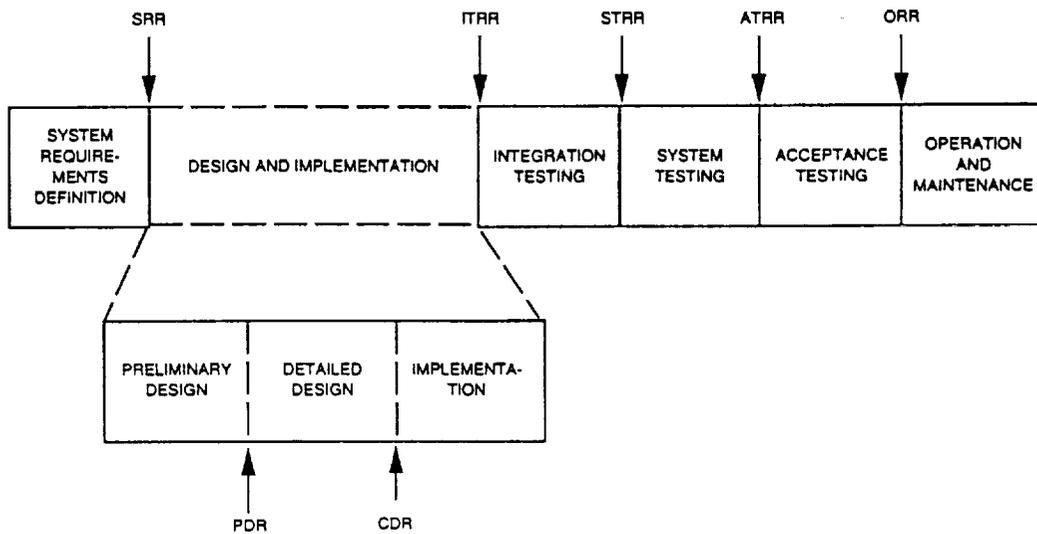
NCC SOFTWARE METRICS DEFINED

On the NCC Project, the metrics program consists of defined, documented processes and measurement tools that provide a quantitative and qualitative representation of the status of a software build or release. Data items including measures of quantity, scheduled and actual progress, number of iterations, and defect information are collected, stored, and reported weekly to provide a snapshot of progress and work yet to be accomplished. All of the tools described in this paper are implemented using a spreadsheet on a 386 or 486 compatible personal computer. The effort expended on project metrics task activities has varied. At peak staffing of the NCC project, 3 full-time analysts were assigned to the metrics activity.

METRICS IN NCCDS SOFTWARE LIFECYCLE

The NCC metrics task originated coincident with the software development activity at the outset of the NCCDS upgrade to support the Second TDRSS Ground Terminal (STGT). It remains a responsibility of the Software Development Department. While the bulk of the data collected and reported by metrics is related to software development, the integration test, system test, and software maintenance tasks are also primary metrics customers. Other project tasks utilize metrics data as an ancillary function.

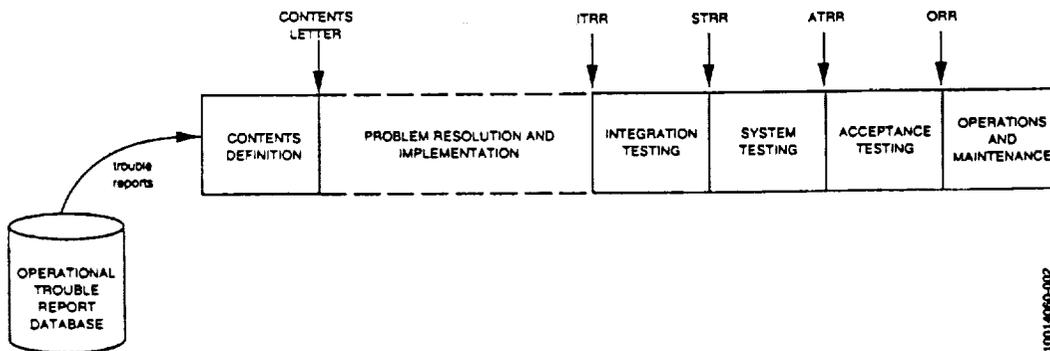
The phases of the NCCDS software development lifecycle are illustrated in Figure 1. The major phases of the lifecycle are requirements identification, design and implementation, and testing. The testing phase is subdivided into integration, system and acceptance testing activities. Each of these phases of testing is performed by a different test group. Milestones that hallmark the software development lifecycle are: system requirements review (SRR), preliminary design review (PDR), critical design review (CDR), integration test readiness review (ITRR), system test readiness review (STRR), acceptance test readiness review (ATRR), and the operations readiness review (ORR). The metrics group provides active support during the design and implementation phase, the integration test phase, and the system test phase.



10014060-002

Figure 1. NCCDS Software Development Lifecycle

The lifecycle for NCCDS software maintenance, illustrated in Figure 2, differs slightly from the above. The major phases can be defined as contents definition, problem resolution and implementation, and testing. The initial phase in the maintenance lifecycle is defined as contents definition as opposed to requirements identification as contents are generally not major system enhancements. They are predominantly fixes to anomalies reported during operational use that have been scheduled for a particular maintenance release. Milestones that hallmark the maintenance lifecycle are an official contents letter, followed by the test readiness reviews listed above. Formal reviews such as SRR, PDR, and CDR are not included in the software maintenance lifecycle.



10014060-002

Figure 2. NCCDS Software Maintenance Lifecycle

METRICS DATA COLLECTION AND REPORTING

When using metrics to support software implementation and testing, the objective is to establish a baseline plan for the work to be done, and then perform work according to the plan. Generally, this approach is the same for development, maintenance and testing with slight variations according to the nature of the task being supported. Data flow between development and maintenance customers and the metrics group is illustrated in Figure 3.

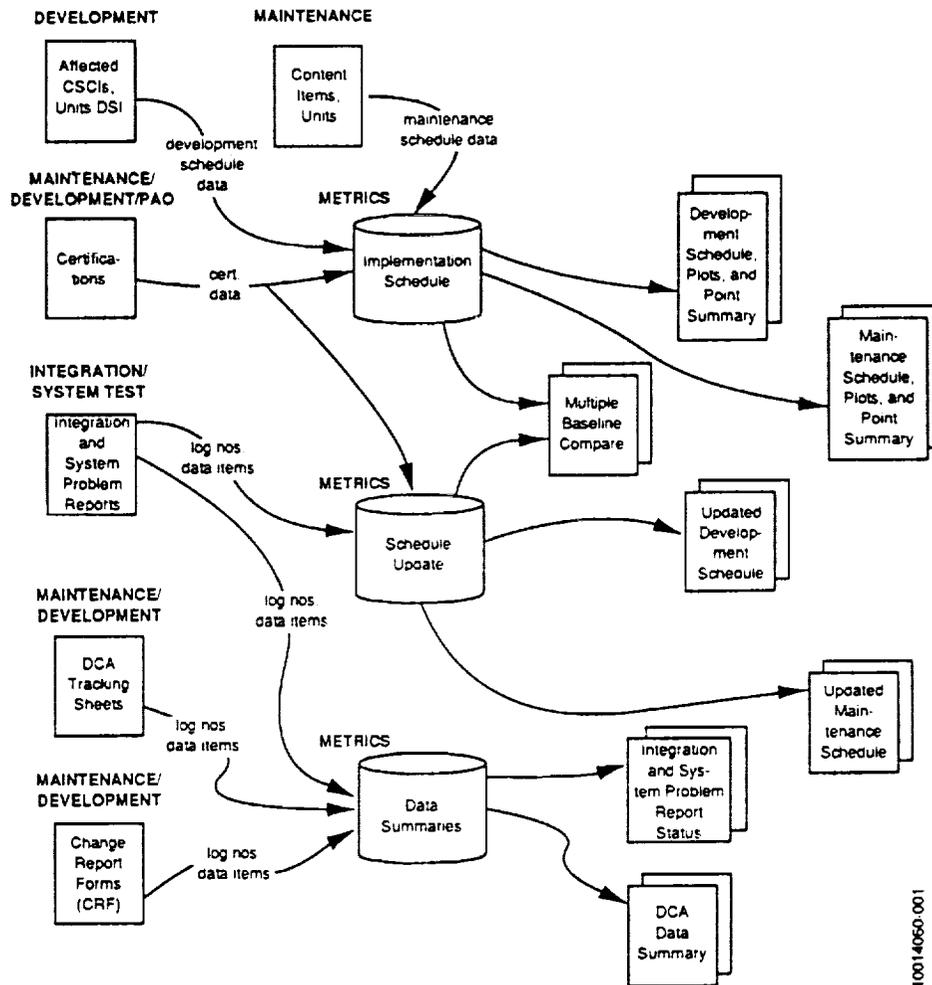


Figure 3. Data Flow - Development, Maintenance and Metrics

Development and Maintenance Support: Implementation Phase

The primary products produced by Metrics during a development or maintenance software implementation are schedules, plots, and points summaries. To begin the process of metrics support of a build or release, both the development and the maintenance organizations must submit lists of the items (units, displays, etc) that are planned for change in the build or release along with an estimate of the size of each item in delivered source instructions (DSI). Development items are identified according to the computer software configuration item and software component (CSCI and SC) along with the item name, and the anticipated impact of the change (a percentage of the DSI.) Maintenance items are identified according to the problem report number, or some other form identifier, along with the item name. This information is used to build an *Implementation Schedule* spreadsheet. A sample portion of a development implementation schedule is shown in Figure 5 below.

IA Number		A										B										C										D									
CSC	SC#	MODULE NAME	OPN DATE	TYPE	IMP	DSI	DESIGN		TEST PLAN		RESP P	RESP I	RECEP	CODE		RESP P	RESP I	RECEP	TEST		RESP P	RESP I	RECEP																		
							Plan	Actual	Plan	Actual				Plan	Actual				Plan	Actual																					
5302	0000	CS1MSG	04/12/93	U	A	96	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CS1MSG	04/12/93	P	C	34	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CS1TAC	04/12/93	U	C	84	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CS1SERVP	04/12/93	U	A	132	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CS1SERVP	04/12/93	U	C	86	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CS1TDRS	04/12/93	U	A	89	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CS1JPM	04/12/93	P	C	17	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CS1VSP	04/12/93	U	A	82	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CTDRSC	04/12/93	P	A	30	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CTDRSF	04/12/93	U	A	146	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CTDRSM	04/12/93	P	C	16	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CTDRST	04/12/93	P	A	12	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CTDRTAB	04/12/93	U	M	42	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CVSPCM	04/12/93	P	C	14	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5302	0000	CVSPCT	04/12/93	P	C	13	06/20/93	06/15/93	06/20/93	06/15/93	D E	M SP	07/01/93	06/30/93	06/15/93	D E	M SP	07/01/93	07/20/93	07/20/93	D E	M SP	07/20/93	07/20/93	D E	M SP															
5304	3000	OBLEDC	04/12/93	P	C	8								04/30/93	04/18/93	A M	D R																								
220		DSI TOTAL		17844	131	131	131	131	131	131	131	131	218	218	131	138																									

Figure 5. Development Implementation Schedule

Included are estimated dates for the design, code and test certification of each item on the schedule. The certification process is defined by the SEAS Software System Development Methodology (SSDM) to record completion of the design, code and test of items in a build or release. Weekly, as the actual certifications are completed, the completion dates are added to the implementation schedule. *Design, Code and Test Plots* are generated that contrast the planned certification progress against the actual certification progress. Each of these graphs also shows growth in the total number of items to be certified. Sample design, code and test plots are shown in Figure 6.

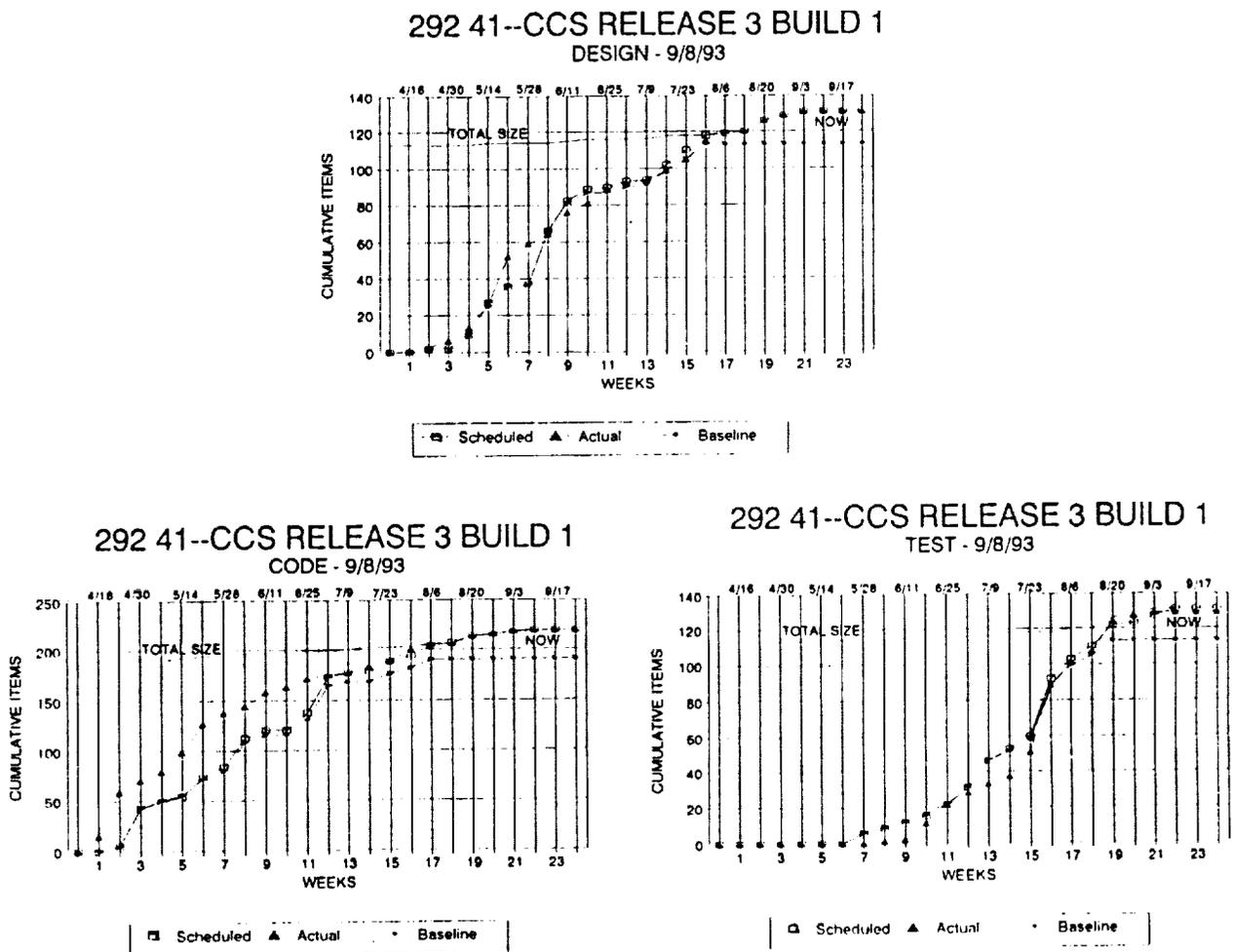


Figure 6. Design, Code and Test Plots

Lastly, a tally of *Points*, Figure 7, is calculated based on the total items to complete and the total completed. Points indicate the amount of credit that can be taken towards the completion of the activity. A different number of points are assigned to each activity - design, code, and test certification - by management. As the size of a build or release grows so does the number of total points. This is reflected in the difference between baseline size and current size, and the points added. Points are calculated weekly and included in the implementation schedule data summary. Each of these products, the schedule, plots, and points summary supports the timely delivery of quality software by providing detailed and high-level feedback on progress and remaining work.

SUMMARY DATA										as of	07/23/93
3329241											
Original Start Date:		04/16/93									
Original End Date:		09/24/93									
Projected End Date:		09/24/93									
Number of weeks elapsed		15 0									
Weeks to Original End:		9 0									
Weeks to Projected End:		9 0									
Activity	Estimated Size			Completed		Yet To Complete		Planned To Be Complete	Percent Ahead of Plan		
	Baseline	Current	Percent Growth	Number	Percent	Number	Percent				
DESIGN	113	118	4 %	100	85 %	18	15 %	118	-15		
TEST PLAN	113	118	4 %	89	75 %	29	25 %	118	-25		
CODE	190	202	6 %	184	91 %	18	9 %	194	-5		
TEST	113	118	4 %	48	41 %	70	59 %	92	-48		
Module Test	0	0	NA %	0	NA %	0	NA %	0	NA		
spare 3	0	0	NA %	0	NA %	0	NA %	0	NA		
spare 2	0	0	NA %	0	NA %	0	NA %	0	NA		
spare 1	0	0	NA %	0	NA %	0	NA %	0	NA		
Total Points	1438	1516	5 %	1232	81 %	284	19 %	1432	-14		

Activity	Points	FISCAL MONTH POINT SUMMARY		CUMULATIVE		%	POINTS Added	
		Month	Planned	Completed	Planned			Completed
DESIGN	4	APR	176	304	176	304	73%	0
TEST PLAN	0	MAY	314	484	490	788	61%	30
CODE	4	JUN	636	336	1126	1124	-0%	24
TEST	2	JUL	306	108	1432	1232	-14%	24
Module Test	0	AUG	84	0	1516	0	0%	0
spare 3	0	SEP	0	0	0	0	0%	0
spare 2	0							
spare 1	0	TOTAL	1516	1232				78
Total	10							

Figure 7. Development Points Summary

Development and Maintenance Support: Testing Phase

At the completion of an implementation phase, the build or release baseline is delivered to Configuration Management (CM) through the Product Assurance Office (PAO) to be placed under configuration control. Once a build or release has been placed under CM control, changes made to the baseline are tracked through the problem reporting system defined in the NCC Standards and Procedures.

The implementation schedule used to account for all items planned for the build or release is now revised to become the *Schedule Update*. The schedule update is used to track the impact of problem report changes on specific items in a build or release as it progresses through testing. The impacting problem report, recertification information, new DSI counts, and variance in the weighted amount of change are recorded on the schedule update. By referencing the schedule update, management and technical staff can identify which software components are being impacted by problem reports and determine if additional build or release items require attention. An enhancement for cross-checking between different builds and releases is the recently developed data summary, the *Multiple Baseline Compare*. The compare shows which software items are being updated simultaneously for different builds or releases. This information is critical in the NCCDS development and maintenance environment to insure the integrity of software products. Implementation activities for different builds and releases sometimes proceed in parallel, requiring that each baseline be updated separately, but yet remain consistent with each other. For example, although a development build may be initiated before a maintenance release, the full development release may not be delivered to the customer until after delivery of the maintenance release. All changes made in the maintenance release must also be present in the superseding development release. The compare report makes the development and maintenance programming staff aware of the need to merge changes from one implementation effort into others as applicable, thus preventing the loss of upgrades and fixes from one baseline to another.

Integration and System Test Metrics Support

There are two levels of testing performed by NCC project staff against each software build or release - integration testing and system testing. The data flow between the test groups and the metrics group is illustrated in Figure 8.

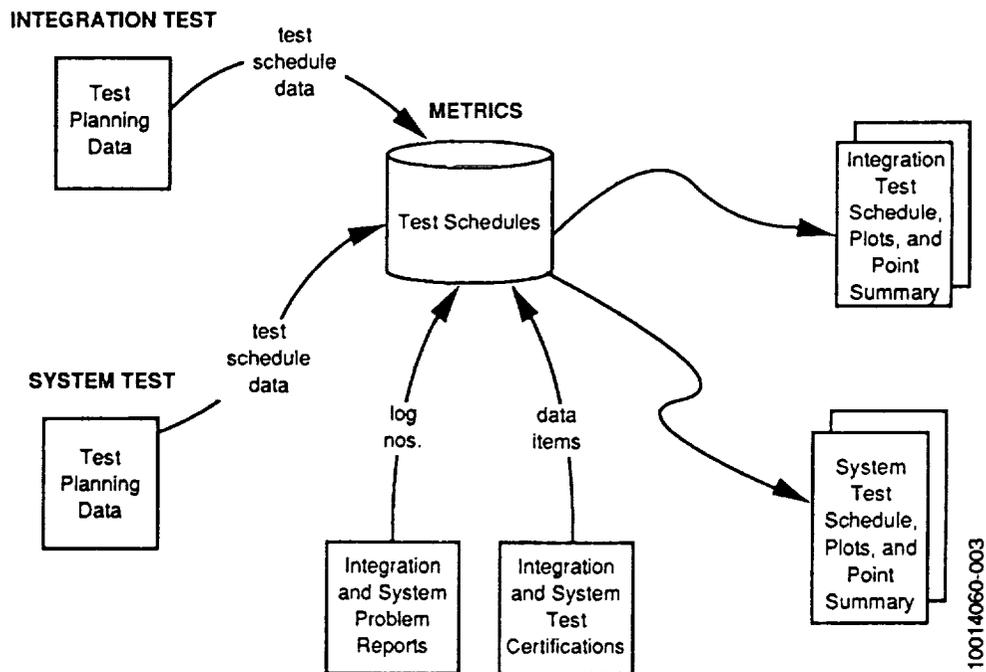


Figure 8. Data Flow - Test and Metrics

As previously stated, at the start of testing the baseline is put under CM control. During integration testing all problems found in the build or release are documented on Integration Software Problem Reports (ISPRs). During system testing, all problems are documented on Software Problem Reports (SPRs). The differentiation distinguishes the test phase in which a problem was found in a build or release.

Similar to the development and maintenance implementation activities, three basic products are provided in support of integration testing and system testing: a schedule of test activities, plots of the progress, and points summary data. Before the start of integration and system test, the integration or system test group provides information on the test cases that will be run against the build or release. Both test groups provide an itemized list with titles and numbers of each test to be run, along with the corresponding CSCI or release requirement. Also provided is additional information including scheduled dates for the start and/or completion of each test, the staff members responsible for each test, and the number of points to be allocated to each test. The use of these tools supports testers and test managers by facilitating the planning of resource allocation for specific tests, identifying problem reports

that are impacting test case progress, and providing weekly feedback on progress. The information collected also helps identify where test procedures are lacking the depth needed to thoroughly test the software.

BENEFITS OF THE NCC METRICS DATA COLLECTION ACTIVITY

NCC metrics reporting makes project status accessible, traceable, and concise. The metrics processes and tools are simple, yet flexible enough to accommodate the specific needs of different managers; the outputs can easily be tailored to each group's needs. Additionally, the use of a project-wide metrics data collection and reporting activity provides an excellent source of information for defect causal analysis. Based on three years of practice, the benefits of the NCC metrics activity can be summarized into three major categories: *Planning*, *Monitoring and Control*, and *Defect Causal Analysis*.

Planning

From the management perspective, the initial and updated schedules provided by the metrics group identify work to be accomplished, in detail, before the start of the effort, for both implementation and testing. Managers are able to establish guidelines for the work to be accomplished during the scheduled interval of time. When used for planning, the implementation and test schedules indicate the concentration of items to be accomplished by date and by functional area. The distribution of staff resources can be mapped and then adjusted as necessary. The use of detailed schedules facilitates the formulation of a workable and realistic plan. From the technical perspective, once the schedules are established they are made available for reference. The plan of action is clear not only to management, but also to those directly responsible for accomplishing the work. Individuals can formulate their own plan for accomplishing work for which they are responsible. Making the schedules available to technical staff also facilitates communication. Each person knows who is responsible for specific items, therefore questions and information can be directed appropriately.

Monitoring and Control

Each manager involved in the completion of an NCC software build or release is required to plan his or her work. Therefore, it is also incumbent upon the managers to compare their planned activities to the progress being made. The points summary and the plots provide at-a-glance feedback on planned versus actual progress. This assists managers in

preparing for monthly status reviews. Metrics reports provide access to historical data that is used as a basis for planning future software implementations.

The regular distribution of metrics reports allows managers and technical staff to identify potential problems as they are developing. It is possible to apply a mitigation action before a problem grows in magnitude. The continuous data capture and reporting cycles facilitate the monitoring and control effort and direct managers to specific areas of concern. Metrics processes track the progress of the implementation down to the unit level. Items that are significantly behind schedule are flagged for further investigation. Similarly, during problem resolution, the progress of test cases and of problem resolution is closely tracked through the data collection and reporting process. The progress of all activities - development, testing and problem resolution - are tied to points summaries and plots. Therefore there are several levels at which information is reported. Plots illustrate progress, and points summaries numerically represent the progress and provide the basis for taking credit for accomplishments on a monthly basis. The schedules contain the detailed information needed by line managers and task leaders.

Defect Causal Analysis

An important initiative in the SEAS program is the defect causal analysis of software implementation and testing efforts. The NCC Project developed a DCA procedure based on data collected by the metrics task, and additional analysis provided by the technical staff. The metrics group provides key data collection and reporting DCA on the NCC project.

In addition to the three basic products already described, the initiation and resolution of ISPRs and SPRs is monitored using the *Detailed Defect Causal Analysis (DCA) Listing*. This spreadsheet and associated plots contain information such as when a problem report was written, the affected NCCDS segment, the manager or task leader assigned to analyze and resolve the problem, and data items that characterize the problem resolution. To aid in DCA, plots are generated that illustrate the characteristics of the problem report resolutions applied to a build or release. When performing DCA of each build or release, it is often helpful to make comparisons of the results against previous build or release statistics. Metrics reports draw attention to areas that are consistently at risk in each subsequent build or release. Results are fed into the subsequent planning process in order to formulate risk mitigation approaches.

An analysis of Release 92.1 statistics by the metrics and development groups for the Service Planning Segment (SPS) of the NCCDS showed that for software items against which formal reviews were conducted by development during the design and code of the release, no software changes were necessary in integration or system testing. As a result of these findings, the SPS group has enhanced its internal review procedures.

PROCESS IMPROVEMENT YIELDS QUALITY IMPROVEMENT

The effect of the project metrics activity on the quality of the NCCDS software is best illustrated in the following chart , Figure 9, that compares size and incidence of errors for four recent NCCDS Releases. Release 90.1 and Release 91.1, software maintenance releases, were implemented and tested prior to the start of the NCC Metrics activity. The metrics activity was initiated with the first build of Release 92.1, a two build software development release. The first maintenance activity to be included in the NCCDS system of metrics was Release 93.1.

On this chart, SPRs are software problem reports written by the NCCDS Project before delivery to the customer, STRs are system trouble reports written by the GFSC acceptance test team. The statistics show that since the advent of the NCC metrics system

development Release 92.1, with the largest number of delivered source instructions, has the lowest overall error rate, and

maintenance Release 93.1, at almost twice the size of Release 91.1, was delivered with half as many total SPRs and STRs.

Release Identifier	90.1 (Maint) Before Metrics	91.1 (Maint) Before Metrics	92.1 (Dev) After Metrics	93.1 (Maint) After Metrics
Release Size (DSI)	7925	18308	112115	36041
Total SPRs Reported	31	135	221	155
Total STRs Reported	15	34	92	25
SPR Errors/KDSI	3.91	7.37	1.97	4.30
STR Errors/KDSI	1.89	1.85	0.82	0.69

Figure 9. Release Size and Error Rate Comparison

Since the system of metrics data collection and reporting has been in use, the rate of errors per thousand DSI has decreased. Collecting defect-related statistics on the schedule update report, the testing schedules and plots, and on the detailed DCA listing has helped to focus attention on critical areas of the software baseline, this has aided in the resolution of problems and the unmasking of additional problems before delivery to the customer. Also, by using the schedules, plots and points summaries to navigate development and test efforts, the NCC Project has met the majority of its internal milestones, and made all of its scheduled deliveries to the customer on time.

SUMMARY

The goal of the NCC Project metrics activity has been to support the project processes and procedures in order that each build or release be delivered on schedule and reflective of high quality. Consistent with this goal, the objectives to be met are to establish plans, monitor the progress according to the plan, and utilize the feedback to effectively manage progress, growth and change during the implementation and test phases. In addition to the above benefits of the metrics data collection and reporting processes, data have been used in the development of Baseline History Reports, and as evidence in internal, division level, and GSFC process audits. Models of the NCC Metrics plan have been used in contract proposals to outline a method for supporting the software development lifecycle.

Comparisons of NCCDS release histories, and the increased level of customer satisfaction have proven that the use of simple tools to support management and technical staff, as described in this paper, have had a measurable effect on the ability of the NCC project to deliver high quality, error-free builds and releases.

REFERENCES

1. Computer Sciences Corporation, *SEAS Software Development Methodology (SSDM)*
2. _____, *Network Control Center Data System (NCCDS) Metrics Handbook*, Draft, September 1993
3. _____, *NCC Standards and Procedures*, Revision 7, March 1993
4. _____, *Release 90.1 Baseline History Report*, February 1991
5. _____, *Release 91.1 Baseline History Report*, January, 1992
6. _____, *Release 92.1 Baseline History Report*, February 1993



NCCDS CONFIGURATION MANAGEMENT PROCESS IMPROVEMENT

Kathy Shay

Computer Sciences Corporation
10110 Aerospace Road, Greentec
Seabrook, Maryland 20706

301-794-1467
301-552-3272 (FAX)

SUMMARY

By concentrating on defining and improving specific Configuration Management (CM) functions, processes, procedures, personnel selection/development, and tools, internal and external customers received improved CM services. Job performance within the section increased in both satisfaction and output. Participation in achieving major improvements has led to the delivery of consistent quality CM products as well as significant decreases in every measured CM metrics category.

GETTING STARTED

In early 1989, the Network Control Center Data System (NCCDS) Configuration Management Section was composed of two full-time technical people, one technical person on loan (to be used as required), one task leader, and the section manager. People had been in these positions for two-three years and knew their jobs. The section manager was new to the company, but not to the CM function, the software/engineering field, nor to Total Quality Management (TQM). The main functions of the CM group are to:

- Provide support to formal project reviews, and baseline and control documentation
- Support configuration item identification and discrepancy reporting system activities
- Maintain software product baselines
- Control changes to various software releases at different testing levels
- Provide status, accounting, reporting, and traceability
- Conduct internal audits and support formal project audits
- Coordinate, track, and report Data Management function activities

The challenge was to "coach" the CM group into one which recognized all of the above responsibilities and responded with quality output to the NCCDS community, consistently.

WHAT WAS IT LIKE BEFORE IMPROVEMENT?

In order to fully appreciate the tremendous gains that have taken place in the CM section, a little time must be devoted to understanding where the section needed improvement. The major areas were:

- o Section Characteristics
- o Procedures
- o Tools
- o Communication

Section Characteristics: There were 3.0 staff to support over 100 people project wide, which produced a total of 450k DSI for the NCCDS system. Although the staff was working in the CM functional area, most were only familiar with the product control aspect. There was no CM status, reports, involvement with the Technical Review Board (TRB) or Configuration Review Board (CRB), no documentation reviews, and no emphasis on quality of work at every level. The task leader was the only person with a college degree and the only person who knew most all machine platforms as well as being able to troubleshoot and analyze CM problems. The task leader was the only person who was cross trained and could step in and help out all areas in addition to helping out during crisis situations. The hours for all personnel were long and frustrating, with little praise for good work. CM had the responsibility to support 7 different software segments (CCS, GNSS, ITS, NFE, NTS, RAP, and SPS), on 4 different hardware platforms (VAX, UNISYS 1100, MASSCOMP, and Intel architecture), in 2 facility areas: The Development Test & Training (DT&T) and Operations. The Section Manager, although experienced and knowledgeable of the CM function, was new to the company and new to the NCCDS. Emphasis on training CM personnel or improving CM processes did not seem to be a priority.

Procedures: Of the 7 segments which CM supported, only 4 systems had any written procedures. Three of these procedure sets were poorly written, incomplete, and incorrect in several areas. The other set of procedures were more of a history of the segment, rather than procedures needed to perform routine functions of that segment. There were few clear steps to follow in any sequential order. Because a new software segment was being developed, there were no procedures in that segment. With no staff assigned to that CM segment on a full time basis, there was little emphasis to write CM procedures for that segment. There were many ideas, troubleshooting mechanisms, tips, procedures, and methods written on sheets of paper gathered in notebooks which tended to be lost easily. The procedures that were documented were inconsistently written across segments. This did not support staff cross training. There was also no

one place which housed all CM procedures. Worse, few people used the correct procedures which did exist.

Tools: Simply stated, most tools which were available to the CM group at that time did not work. Custom made tool sets were not maintained thereby causing errors when unknowing staff used them. There was no time scheduled to investigate the root causes and correct problems, just time to fix them. There were many laborious work-arounds that staff used because automated routines were not available or the ability to keep them current did not exist. The inefficiencies resulted in long processing times, incorrect output, and longer fix times. The mere difficulty in using some of the tools themselves caused errors. These internal CM problems were having enormous effects on the rest of the project, in terms of schedule, reliability, cost, causing staff frustration and lowering confidence in the ability of CM to do the job.

Communication: During this time, CM processing time requirements were not recognized on any official project schedules. The time CM required was discussed in management meetings, although internal schedules never reflected the resource. The section manager discussed with the development and test managers the need to "steal" a day on each end of "their" schedules to accommodate CM requirements. This method of acquiring schedule was not conducive to smooth transitions. Most times, the software deliveries were made on the last day of their schedule at 6:00pm and test expected to start the next day at 6:00am. There was no routine status accounting or reporting to the project of CM units processed, reports tracking documentation, or CM efficiency and productivity. In addition, there was little input from CM to the overall project planning process, needs, and problem areas. Participation in CSC Project Management System (PMS) planning, weekly reporting of CM activities to the Assistant Technical Representatives (ATRs), and monthly presentations to GSFC management of CM accomplishments/problems areas was weak.

WHAT WAS DONE?

There were two efforts undertaken to improve the CM function: 1) A management initiative to improve section processes and routine ways of conducting business, and 2) Establishment of process improvements through the participation of CM team members in the Task Oriented Process Improvement Committee (TOPIC).

MANAGEMENT INITIATIVES: After assessing the situation, management devoted emphasis to 1) staffing 2) project participation 3) defining procedures 4) improving tools 5) providing status and reports, and 6) self evaluation of CM processes. Several areas were totally re-engineered.

To remedy the staffing situation, over the next year and a half, there were 8-9 staff personnel hired to work in the CM function. An additional person was on loan, part time, to assist with tools and CM sponsored a summer hire, who helped with the CM data base development. At the end of the CM personnel transition, all personnel had completed their bachelors, 3 people had completed their masters, 3 people were working on their masters, and 1 person was working on a PHd. This higher level of educated personnel was then applied to every segment, which allowed a different degree of work to be performed. In reality, this transition of personnel took nearly 2 years to evolve, and has never stopped. The increased capabilities of the personnel have allowed a much easier cross training of different personnel on different software segments. It reinforced the necessity to have educated and trained personnel and precipitated regular training for CM including internal classes, SEAS courses, vendor classes (both brought into CSC and attendance on vendor sites), and attendance at conferences. The higher level of personnel expertise enabled CM to be able to analyze, troubleshoot, and resolve problems within our own section. People who were doing the work and making errors were able to begin fixing them. This also enabled the group to have insight into what and where some of the root causes of the problems were.

As CM began working more with project management, quality assurance, release leaders, and other technical people, the need for CM to identify processing time on schedules became a reality. Internal schedules contained references to CM time required as well as providing detailed planning schedules prepared by release leaders used to plan Integration, System Test, Acceptance Test, and Operations transitions. CM personnel were able to plan for work and knew what the project deadlines were and where CM fit into the big picture. CM also began to schedule machine/facility resources in the Software Development Facility (SDF), the DT&T, Emergency NCC (ENCC), and Operations areas. By this time the NCCDS had added two facilities; one a development facility in the CSC Greentec I area, and the other an ENCC at the GSFC facility. This added to the CM responsibility of maintaining equal configurations for each release. Having to maintain multiple releases at different test levels sometimes necessitated that CM have their own machine time to perform some processing and installation functions. CM therefore began to schedule resources in the required facilities and "piggy-backed" off other's scheduled time when there was no CM or other function impact.

Procedures were another area which improved dramatically. Personnel have documented or updated all CM procedures. Procedure formats were standardized across all segments in a logical step by step fashion. They were also written to be user friendly, incorporating helpful processing notes. Today the

procedures are used as a training tool for new CM personnel. They are also updated on a routine basis, as a part of the CSC Program Management System (PMS) accountability system. In addition to the documented detailed procedures, an NCCDS CM Software Plan was developed and is revised periodically.

Two major efforts were undertaken to improve CM tools used on the CCS and SPS segments.

1. In 1989 a study was undertaken to determine the best CM tool to use on the VAX. After the investigation was complete a presentation and report was provided to GSFC. The decision was to continue to use the existing CSC written tool which would be enhanced and coupled with Digital Equipment Corporations MMS (Module Management System) tool. Task personnel basically completed the VAX tool effort in 1990. Upgrades have been added each year to continue improving tool productivity and efficiency.

2. Another effort was undertaken in 1990 to improve the Unisys tool. Although there were off-the-shelf tools evaluated, none provided the control, reporting, and speed that were desired. Over a period of one year, task personnel first identified the areas which required immediate attention and made the proper fixes. Second, desired enhancements were identified and gradually added. Both the fixes and enhancements were applied in an internal controlled manner. Internal Problem Reports were written and resolution

recommendations were evaluated. Various fixes and enhancements were packaged together and released in builds to the tool. The build was first tested in a testbed on the SDF Unisys. After all bugs had been eliminated, it was inserted into the regular controlled tool which was used in the SDF. Figure 1 contains a high level overview of the SPS tool fixes and enhancements. A summary was presented to the project CRB, approved and then applied to the DT&T controlled version of the tool. These enhancements allowed the tool to run more efficiently and later several utilities were automated into a menu program.

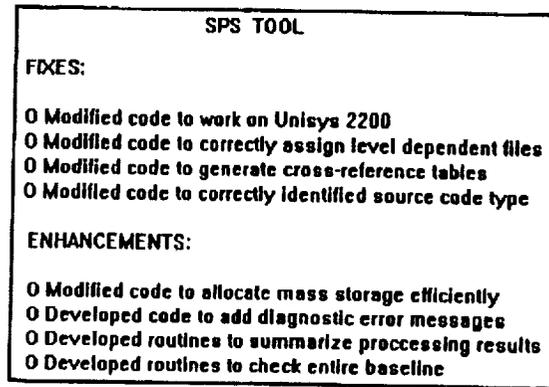


Figure 1

The benefits as a result of the tool improvements have been noticeable throughout the section and the project. No longer are incorrect software versions of a unit delivered to test. Listings are routinely run, reviewed, and archived for traceability purposes. These are used later for troubleshooting, if required. The two-three day test sessions have not been hindered by the inability of CM to locate an incorrectly assigned level dependent file. Should there be a problem, listings with diagnostic error messages assist in locating the source of the problem quickly and easily.

93.1 SPRs at LVL 2 System Test WEEKLY FROM 12/04/92 - 4/30/93

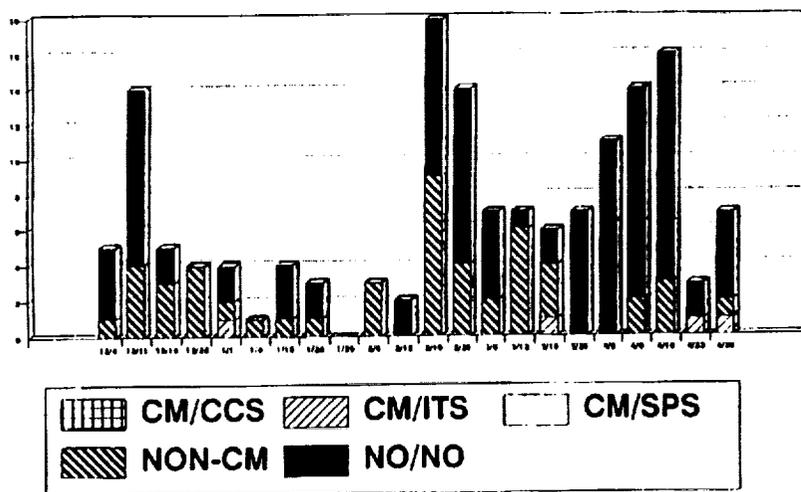


Figure 2

Another tool enhancement to the section was the creation of a CM relational data base to provide traceability, reports, and on line information. Up to this time there was no routine accountability of CM activities to the project. There were no reports or statistics kept within the section. Task personnel developed a data base to house CM

data from the Internal Software Delivery Forms (ISDF) and data regarding the CM processing. Data such as segment, subsystem, type of delivery, ISPR/SPR/STR number, unit name, type of unit, date received, and date processed was collected and entered. A units processed report for each release is provided to the project each week. Another report showing elapsed time indicates CM efficiency in processing deliveries from the time of receipt to the time available at any level for testing. Graphs are produced at each project phase for each release and show what (if any) CM problems are occurring on each segment. Figure 2 shows the CM problems by segment for Release 93.1 during the System Test phase. These weekly graphs act as a catalyst for internal CM Defect Causal Analysis (DCA) and in continuing process improvement. This database eventually was merged with the System Engineering Project Database (SEDB) and is known today as the Configuration Management Database (CMDB) and is maintained and has been improved by the System Engineering database section.

CM has code counter tools for each of the NCCDS segments. Over the last two

years, each tool has been upgraded to be more efficient, easier to use, and has been fixed to reduce errors. All of the tool improvements have enabled the CM section to provide products to both internal and external customers faster, better, cheaper, and more accurately.

THE CM TASK ORIENTED PROCESS IMPROVEMENT COMMITTEE (TOPIC): At the same time the above management initiatives were taking place, the CM group began setting aside a small amount of time each week to discuss changes in the section that would improve quality, productivity, worker satisfaction, and reduce errors. In 1990, with CSCs increased interest in Total Quality Management, the group became known as the Task Oriented Process Improvement Committee (TOPIC). The section manager sponsored the group, and every six-eight months the group chose a new facilitator.

The first project the group undertook was to design a set of checklists which were to be used as a verification tool and used in conjunction with the processing procedures. Although the processing procedures were being revised, the group wanted a high level composite list of "musts" that should be accomplished that a second person could verify to ensure the proper steps had been followed. Each segment lead prepared a set of "checks" for each segment and each processing phase. For example, the CCS checklist for processing a delivery includes seven "checks". The verifier completes the checklist as the actual item is checked. The verification is a combination of checking on the terminals and checking the printouts.

The checklists can be used for test levels 1, 2, and 3 and are signed and dated by the verifier. An example of an SPS checklist is shown in Figure 3. Other checklists have been designed for phases such as: 1) Creating a Baseline, 2) Processing a Delivery, 3) SDF/DDT Transfer Tape Update, 4) Creating Failover Tapes, 5) Installing a Release, 6) Making Operational Tapes, 7) Processing Symbolics, Procs, Schema, Templates, Maps and QLP Reports, 8) Updating a Baseline, 9) Installing a Release Into OPS, and

SPS VERIFICATION CHECKLIST PROCESSING SYMBOLICS, PROCS AND SCHEMA					
Release#:	_____	Log Number:	_____		
Installer:	_____	Hardware:	_____		
Date:	_____	Directory File:	_____		
		ISPR/SFR/STR:	_____		
L1	___	Verify output listing from the @DELIVERY execution for a successful copy to level 3 and current date for the association CH baseline file.			
L1	___	Verify compile table (UCL.COMPILE/ACS) to ensure all association of the delivered of the delivered log number(s) in the file.			
L2	___				
L3	___				
L1	___	Verify under 65K compile file (COMPS) for delivered units.			
L1	___	Verify over 65K compile file (COMPSO) for delivered units.			
L1	___	Verify map file (MAPS) for associated transactions.			
L2	___				
L3	___				
L1	___	Verify output listing from @COMP-CHECK execution for over and under compile errors.			
L1	___	Verify output listing from @MAP-CHECK execution for map errors.			
L2	___				
L3	___				
L1	___	Verify that all delivery letters and output listings are dated/initialized and placed in proper baseline binders.			
L2	___				
L3	___				
L1	___	Verify that the CH database has been updated.			
L2	___				
L3	___				
COMMENTS/PROBLEMS:					
LOG NUMBER	LEVEL-1 VERIFIER	LEVEL-1 REVIEWER	LEVEL-2 VERIFIER	LEVEL-2 REVIEWER	LEVEL-3 VERIFIER
___	___	___	___	___	___
___	___	___	___	___	___

Figure 3

10) Deleting Units. The checklists are tailored to incorporate segment differences.

Another process established to be used with the checklists, was the incorporation of a reviewer. Usually, the reviewer is either the task leader or the section manager. The review session takes place prior to the delivery or product being installed in the SDF. The processor, verifier, and reviewer go over the delivery from beginning to end to ensure all steps have been completed properly and without error. All printouts, listings, checklists, and original delivery paperwork are reviewed and retained by the CM lead for the segment. Any future inquiries into a delivery, can be recovered and investigated if required. This

- CM TOPIC ACHIEVEMENTS**
- Standard Verification Check Lists
 - Standard Code Count Form
 - Computer Time Usage Log Form
 - Operational Software Installation Form (OSIF)
 - Monthly Backup Forms
 - Delivery Form Error Reductions

Figure 4

three-pronged approach has added discipline to the overall process and assisted greatly in the reduction of errors. A partial list of CM TOPIC achievements is shown in Figure 4.

The TOPIC also initiated what they called a "shake down" test. CM processes all the deliveries for a given release for a particular group (i.e Integration Test, System Test, or Acceptance Test) to begin testing. After the installation is made on either the SDF or the DT&T machines, but prior to turnover to the test group, CM coordinates a "shake down" test. This test is a multi discipline team effort comprising the CM segment leads, an integration tester or system tester, a computer operator, maintenance and data base personnel. The CCS, ITS, and SPS segments are brought up, connected, and are checked to ensure all segments "talk" to each other. Although no functions are performed, data passed, or test cases run, this simple check has pinpointed several errors. These problems were cleared up prior to the baseline being provided to the "internal test customer". Time is saved by CM and the testing groups and customer satisfaction is enhanced.

WHAT WAS IT LIKE AFTER IMPROVEMENTS?

Specific: There have been many improvements over the past three years. In addition to the overall management initiatives and TOPIC achievements there have been other individual task improvements. Listed below are only those specific improvements which have been documented through either the CSC Code 550 or 530

cost avoidance system:

IMPROVEMENT	STAFF HOURS-\$ SAVINGS/YEAR
1. Revised NTS Build Procedure	- 24 staff hours - 5500 sheets of treated paper per year
2. Improved CM Procedures	- 60 staff hours
3. Simplified Delivery Process	- 32 staff hours
4. Modified CCS Compilation Process	- 150 staff hours - 18 VAX CPU hours in 1991 98 VAX CPU hours in 1992 106 VAX CPU hours in 1993
5. Designed Standard DSI Form	- 114 staff hours - \$ 14,022 over 3 years
6. Eliminated Duplicate DSI Counts and Reduced Errors	- \$ 5,530 over 3 years
7. Revised and Designed New CM Valtab Procedures and Form	- 26 staff hours

Statistical: Because either little or partial data was maintained in the late 80's in the CM section, the best possible attempt has been made to present fair and accurate data. Emphasis has been placed over the last few years on quality deliveries to our internal customers. Some of those internal customers are Integration and System Test.

During the Integration Testing phase, Integration Software Problem Reports (ISPRs) are written to document problems. Available data show there was a 50% reduction in errors over the three builds after the 89.1 Release series. From Release 90.1 to Release 93.1, CM ISPR errors decreased to around 11%. During the Release 3 Build 0 integration testing, there were zero CM errors out of a total of 51 problems.

CM was accountable for nearly 20% of the project's Software Problem Reports (SPRs), written during System Test phase, up through the 89.1 Release series. Figure 5 shows the SPR trend for earlier releases. During the time many of the

improvements were initiated, Release 90.1 was being developed and CM SPRs fell to about 13%. This trend continued into the development of Release 91.1A, and CM SPRs dropped to under 6%. By the time Release 92.1 was being System Tested, CM SPRs were down to just over 4%.

More recently, as shown in Figure 6, the trend has continued to be the same. Release 93.0 found CM SPRs at zero. Release 93.1 was a much larger

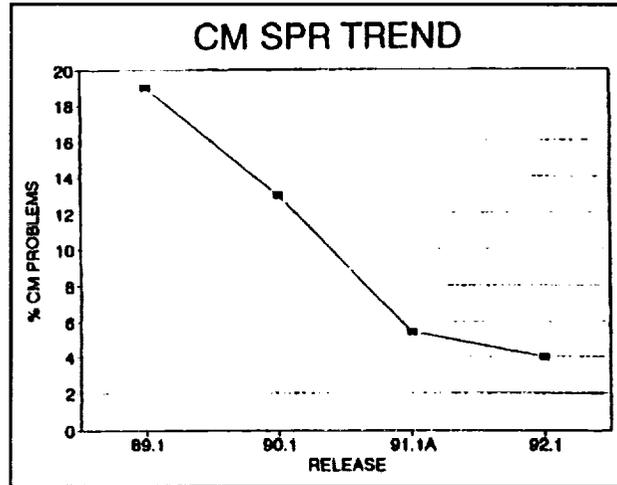


Figure 5

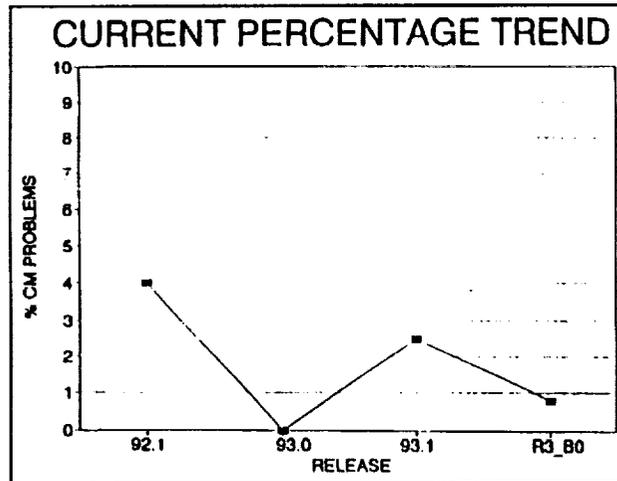


Figure 6

development effort and CM SPRs were around 2.5%. Release 3 Build 0 is currently under test and to date, CM SPRs are .8%. Clearly, the number of errors attributable to CM has decreased substantially. Obviously, the time spent in correcting problems has decreased accordingly, and a much greater confidence level has been achieved from groups receiving CM products. The project can now count on CM to make internal schedules.

Figure 7 shows a composite of the number of total problems against the number of CM problems.

This improvement has also been seen in installations and products delivered to GSFC. During the Release 89.1 series of deliveries to Acceptance Test, CM accounted for over 13% of the errors identified in the release. Acceptance Test documents problems on a System Trouble Report (STR). Over the next three releases, problems

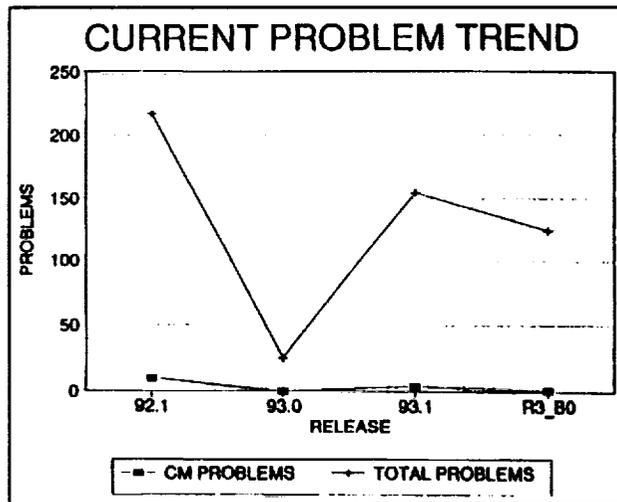


Figure 7

attributable to CM fell to less than 2.5%. There were also no CM STRs for Release 93.0 and only one CM STR for Release 93.1.1, which went operational recently. The turn around in the percentage of CM problems has been dramatic over the past few years. The quality and dependability of CM products and services to our customers has been increased by these measurable results.

Participation: Key to the improvements has been the acceptance of all CM team members to want to make a difference and to make things better. Early in the process, team members recognized the need to become more efficient and more productive. As the opportunity became available to participate in TQM committees all CM section members took advantage. The contributions to the CM TOPIC through the years has been directly responsible for many of the CM successes and improvements. CM has had 100% participation in the five major TQM committees. Two of the five first committees were facilitated by CM members. All CM team members have been involved in Process Improvement Committee (PIC) Process Action Teams (PATs). This participation across project functions to improve a process has provided team members with insight into resolving multi disciplined problems which benefit everyone. The enthusiasm and willingness of CM team members to participate at all levels of TQM activities has strengthened the project, the section, and the individuals involved. Everyone wins.

Recognition: When a job needs to be done, it should not be done to seek recognition. Over time, as each year rendered better results, individuals within the CM group and the team as a whole realized technical and professional recognition. Listed below are some of those achievements:

- o Documented and received four Flight Dynamics Quality Improvement Ledgers citing success stories
- o Many of the CM achievements have been publicized in the SEAS Total Quality Management Highlights
- o Individual CM members and the CM team have received NCC Awards and Recognition Committee (ARC) monthly recognition certificates
- o Documented and received three recent cost avoidance success reports
- o Two CM team members received FDTG Engineering Employee of the Year Awards
- o One CM team member was honored as the first recipient of the NCC Project Dedication, Adaptability, Team Spirit, Unique Solutions and Motivation (DATUM) Award
- o One CM team member was a winner of the SEAS TQM Involvement Award

The team was also nominated for the 1992 SEAS Quality Service Award and an individual nominated for the 1993 NTG Quality Service Award. In addition, there

have been letters of commendation from other CSC codes and from the GSFC customer on CM team members excellent service and support.

CONCLUSION

This paper has listed many CM improvements over a wide spectrum and shown meaningful statistical evidence of positive results. The above findings, however, do not mean the group is perfect or that the job is done. The challenge is to provide "continuous" improvements. Because the gap has been tremendously narrowed, future improvements will probably not be measured in whole percentages. The hard job will be to continue to chip away until the goal is obtained. The goal is to have zero processing errors, to provide internal and external customers CM products and services which are error free, and to continue to increase CM efficiency and productivity. "In CM, we don't make the software... we make it better!"

PROBLEM REPORTING MANAGEMENT SYSTEM PERFORMANCE SIMULATION

Author: David S. Van Natta, Raytheon Service Company

SUMMARY

This paper proposes the Problem Reporting Management System (PRMS) model as an effective discrete simulation tool that determines the risks involved during the development phase of a Trouble Tracking Reporting Data Base replacement system. The model considers the type of equipment and networks which will be used in the replacement system as well as varying user loads, size of the database, and expected operational availability. The paper discusses the dynamics, stability, and application of the PRMS and addresses suggested concepts to enhance the service performance and enrich them.

INTRODUCTION

The purpose of the PRMS simulation is to develop an understanding of potential system behavior, with the goal of using that understanding to make design decisions involving the system. The process of developing and running the simulation involves some combination of the following activities; preliminary analysis of proposed system design; model design and coding; verification; validation; experimental design; performing simulation runs to produce output data; and statistical analysis of output data to estimate parameters. Each of these activities contributes to an understanding of the system behavior.

The intent of this paper is to describe the performance model of the PRMS. Performance modeling is a technique that employs classical operations research and simulation to quantify process improvements and expected operations performance in the spirit of Total Quality

Management Continuous Improvement. As a TQM tool, performance modeling is a natural extension of the flow chart, the histogram, the cause and effect ("fish bone") diagram, and other statistical process control tools. Performance modeling allows managers to both visualize and quantify a process, and to project what effects changes will have on that process.

MODEL SPECIFICATIONS

Use of NASAs Trouble Tracking Reporting Data Base (TTRDB) has increased significantly over the past few years. Reporting requirements provided by the TTRDB no longer satisfy the increased use of the system which currently serves the Space Network (SN). As a result, a replacement system, PRMS is being developed to replace the TTRDB. PRMS will support and provide enhanced reporting capabilities to the SN and the Ground Network (GN). Notwithstanding, the PRMS will accommodate reporting capabilities for the Second TDRSS Ground Terminal (STGT), The White Sands Ground Terminal Upgrade (WSGTU), the Gamma Ray Observatory Remote Terminal System (GRTS) and four TDRSS operations. Current reporting capabilities are not configured to handle report inputs from STGT and WSGTU site configurations or for more than three TDRSS operations. The PRMS will also contain the required provisions for maintaining problem reports on ground sites supporting the Network Control Center (NCC) scheduled missions (Elwell, 1993).

The design of the PRMS is portrayed as a single server unit servicing multiple users in multiple locations. The functions of the system are to provide:

- Direct on-line entry, by NCC operations personnel,
- User logon/access security,
- Multilevel access characterized by user access security,
- Verifiable Data entry,
- Graphical User Interface,

PRMS Database and Network Connection

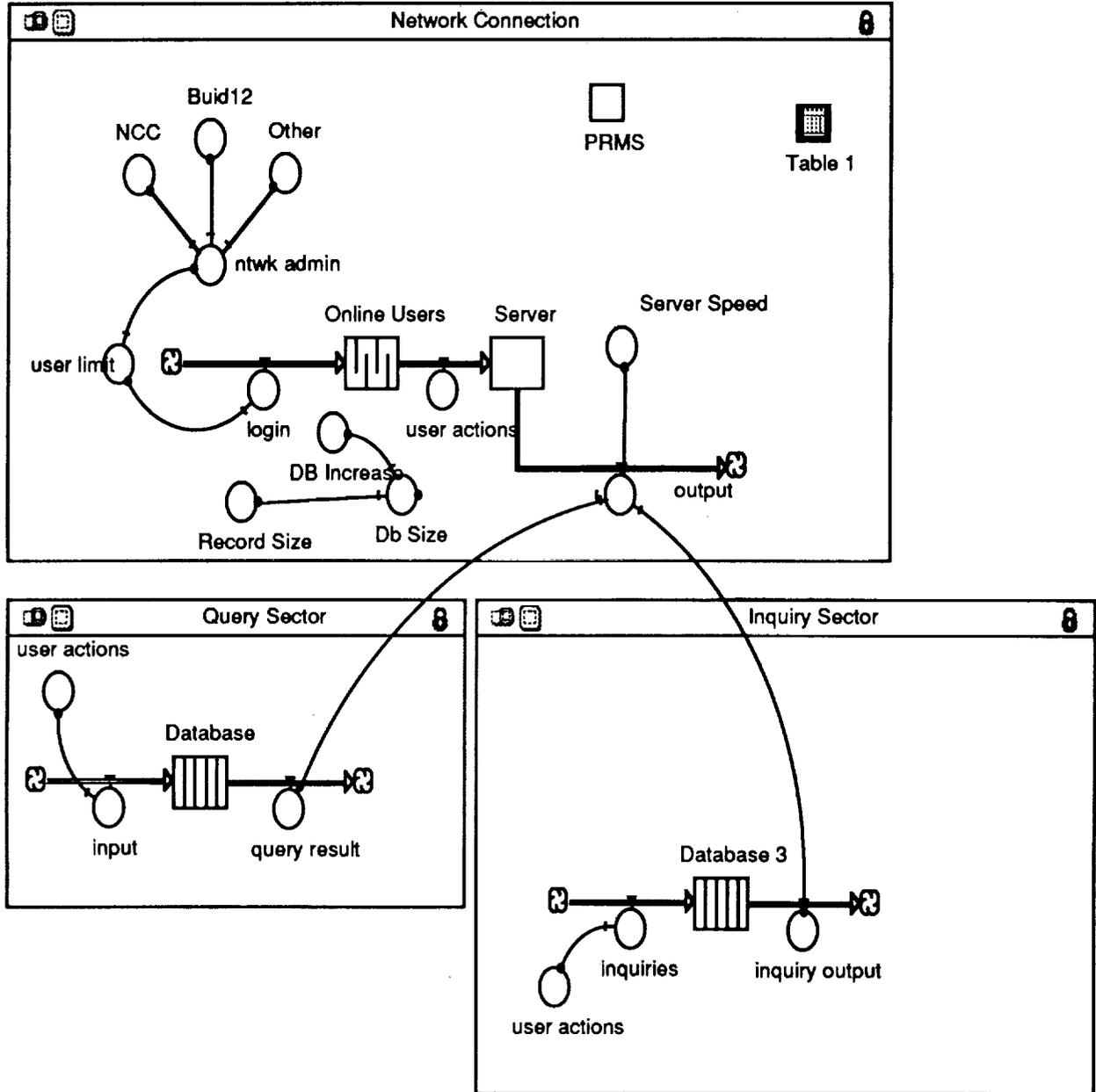


FIGURE 1. PRMS SIMULATION STRUCTURE AND NETWORK CONNECTIONS

- Capability of building queries, generating output reports, batch processing, sorting, indexing, consolidation of existing reports, slaving reports to a master report, electronic approval, graphical output, and auto-numbering,
- Editing, appending, and viewing of reports, and
- Network capabilities.

MODEL ORGANIZATION

A software organization was developed for implementation of an animated simulation model based on the previous information and forms of animated even. It includes a simulation model, a static background, static elements, a dynamic foreground, dynamic actors, and the trace file.

Figure 1 represents the basic simulation structure and network connections.

The simulation contains some initial sets of logical processes. Ideally, these processes should be distributed accross the system so that response time delays are minimized. Thus, the simulation model becomes a surrogate for actually being able to experiment with the yet to be built PRMS. Since random samples from input probability distributions combined with current usage rates of the TTRDB are used to “drive” this simulations model, basic simulation output data or an estimated performance measure computed from them are also random.

Because of the random nature of the simulation input, the PRMS model produces a statistical estimate of the (true) performance measure, not the measure itself (Bartley, Fox, and Schrage, 1987) .In order to predict statistically precise estimates that are free of bias;

- The length of run was set at 1000 time units
- The number of independent simulation runs was set at 5, and
- No simulation warm-up period was needed.

The independent runs were made for each configuration of the system (386 based, 486 based, Macintosh based, etc.), and the average of the estimated performance measures from the individual runs was used as the overall estimate of the performance measure. Independent runs mean using different random numbers for each run, starting each run in the same initial state, and resetting the model's statistical counters back to zero at the beginning of each run.

EXPERIMENTS

The experimental study of the PRMS used three type of servers, a 486/33, a 486/66, and a Macintosh 840A. In all cases, a user action that arrives at the queue is sequentially served and is thereafter routed to either an input or an output function with a predetermined probability. The service time of a job at the server is generated from a negative exponential distribution in which all servers are assumed to have an identical mean service time related to their clock rates. The variants were the type of servers and the increase in user rates vs. data base size. An example graphic output is shown in Figure 2.

The simulation was run for each of the server types with varying user loads and data base usage rates. Steady state analytical models were used for comparison. (See Kleinrock, 1976 for a discussion on steady state analytical queuing models associated with computer and communications network performance issues.) The dynamic outputs, both numerical and visual, are being evaluated for performance vs. cost concerns.

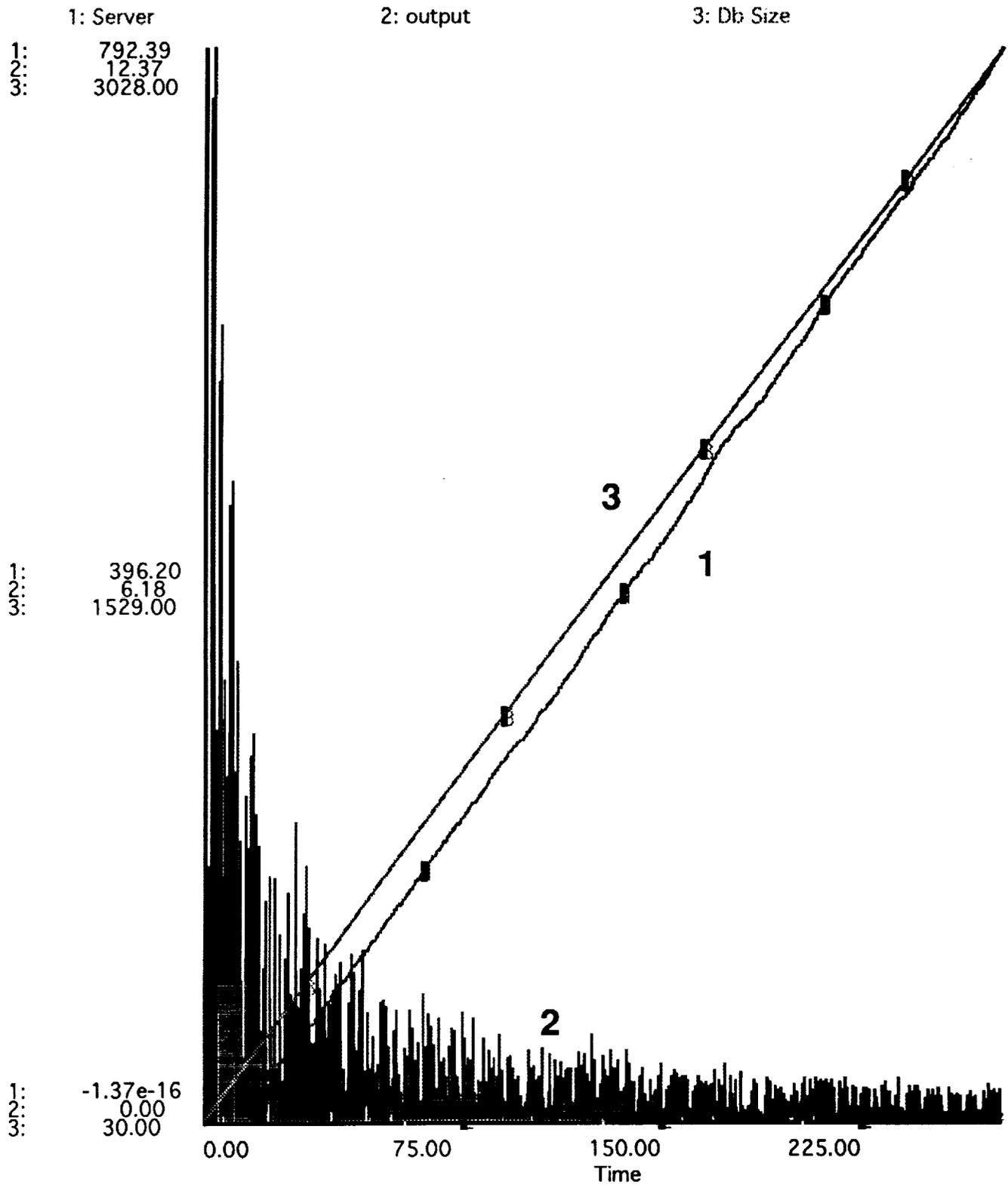


FIGURE 2. GRAPH OF OUTPUT AS A FUNCTION OF DATA BASE SIZE AND SERVER LOAD

CONCLUSIONS

The PRMS is still in the design phase. This model is being used to provide quantitative performance measures to substantiate a specific system platform to build on. The model will be expanded to include system maintenance and a more refined data base input mechanism as the design progresses. The simulations model has demonstrated that a number of optimizations can be performed transparently by a runtime model and that an accurate depiction of the PRMS can be made in its various states of design.

REFERENCES

- Bratley, P., B. L. Fox, and L. E. Schrage. 1987. *A Guide to Simulation*, New York, New York: Springer-Verlag
- Elwell, D. 1993. *Problem Reporting Management System Operations Concept*, Greenbelt, Md: Goodard Space Flight Center
- Kleinrock, L. 1976. *Queing Systems*. Volume 2: Computer Applications. New York: Wiley.



SESSION 2: NETWORK OPERATIONS

1

2

3

4

5

6

7

8

Enhanced Networks Operations Using the X Window System

Irving Linares

NASA Goddard Space Flight Center
Network Control Systems Branch
Greenbelt, MD 20771

Keywords: X Window, GUI, Client/Server, Open System Architectures, Multimedia

Summary

We propose an X Window Graphical User's Interface (GUI) which is tailored to the operations of the NASA Goddard Space Flight Center's Network Control Center (NCC), the NASA Ground Terminal (NGT), the White Sands Ground Terminal (WSGT) and the Second Tracking and Data Relay Satellite System (TDRSS) Ground Terminal (STGT). The proposed GUI can also be easily extended to other Ground Network (GN) Tracking Stations due to its standardized nature.

1 Introduction

Currently, the operators of the various heterogeneous computer systems at the NCC, NGT, WSGT and the future STGT need to monitor and control the computer systems under their responsibility using dedicated terminals for each system. Some of these terminals are just ASCII interfaces which provide no graphical capabilities, thus they can not support intuitive user-friendly Graphical User Interfaces. This makes performance data analysis unnecessarily harder than what it should be. Some operators even have to alternate between different locations within the control room to perform status and command operations.

With the increased availability and affordability of open networking among major workstation vendors, it is a sensible solution to integrate multiple status and control

terminals into a reduced number of terminals with multiple windows. The X Window System[1] was just designed to meet this requirement. A multiple window display offers a much broader centralized picture of the overall system status while it could simultaneously improve the efficiency of the control center operations. It can also provide an intuitive user-friendly graphical model of the system interfaces that can be designed to minimize entry errors.

Many other Commercial-Off-The-Shelf and NASA custom-developed services can be easily incorporated into the basic X Window System to dramatically improve the *intercenter* and *intracenter* Human-Computer and Operator-Operator Interfaces once the basic infrastructure is in place[2]. These services, which could be encompassed by the *multimedia* term include but are not limited to: FAX, NASA Select Television, enhanced electronic mail containing text, graphics, animated sequences, images and audio), Video Teleconferencing, Space Tracking and Data Network (STDN) Electronic Library with hyperlinks to engineering drawings and technical manuals, CD-ROM interactive training, *what you see is what you get* (WYSIWYG) word processing, still image capture and display, Logistics, and Facilities Automation among others.

A well designed distributed system should provide fast real-time response, high availability and reliability, security, robustness, fast prototyping, openness in terms of multivendor support and transparent structured transitioning into operational systems. The UNIX Operating System in conjunction with the X Window System can satisfy all of the above requirements. NASA can advantageously use the X Window System to integrate already developed standalone applications to provide all the aforementioned services in a cost-efficient dependable way while still satisfying all the operational and real-time constraints of a control center environment. The X Window System application proposed in this paper, with its high degree of modularity, multi-vendor compatibility, flexibility and intrinsic expandability should provide a long lasting investment to safely launch us into the next century control center.

2 Proposed System Architecture

The proposed baseline system standardizes on well accepted International and *de-facto* client/server models. The basic building blocks are the Open System Interconnection (OSI) Seven-layer Reference Model, the Transmission Control Protocol/Internet Protocol (TCP/IP), the Integrated Services Digital Network (ISDN) Protocol and the MIT X Window X11R5 (or later) System. Additionally it is suggested to adopt the Motif Window Manager (mwm) and the UNIX Operating System which are supported by all the manufacturers under the current NASA Scientific and Engineering Workstation Procurement (SEWP) contract[3]. Another requirement is RGB video monitors and RGB video switching. All the displays under the SEWP also are compliant with RGB signaling. A typical configuration is shown in Figure 1.

We specifically propose to separate video and voice from the data on the network to further enhance the robustness during possible emergencies and to prevent LAN overloading under normal operating conditions. In the future, if the Fiber Distributed Data Interface (FDDI) becomes commonplace, we *may* consider mixing voice, data and video on the same network.

3 Functional Requirements

We propose that minimally, an *administrative* workstation should have access to the following services:

- Technical Information Program (TIP)
- Automated Logistic System (ALS)
- Enhanced Logistics Information Management Systems (E-LIMS)
- Interactive Multi-Media/Computer Based Training (IMM/CBT)

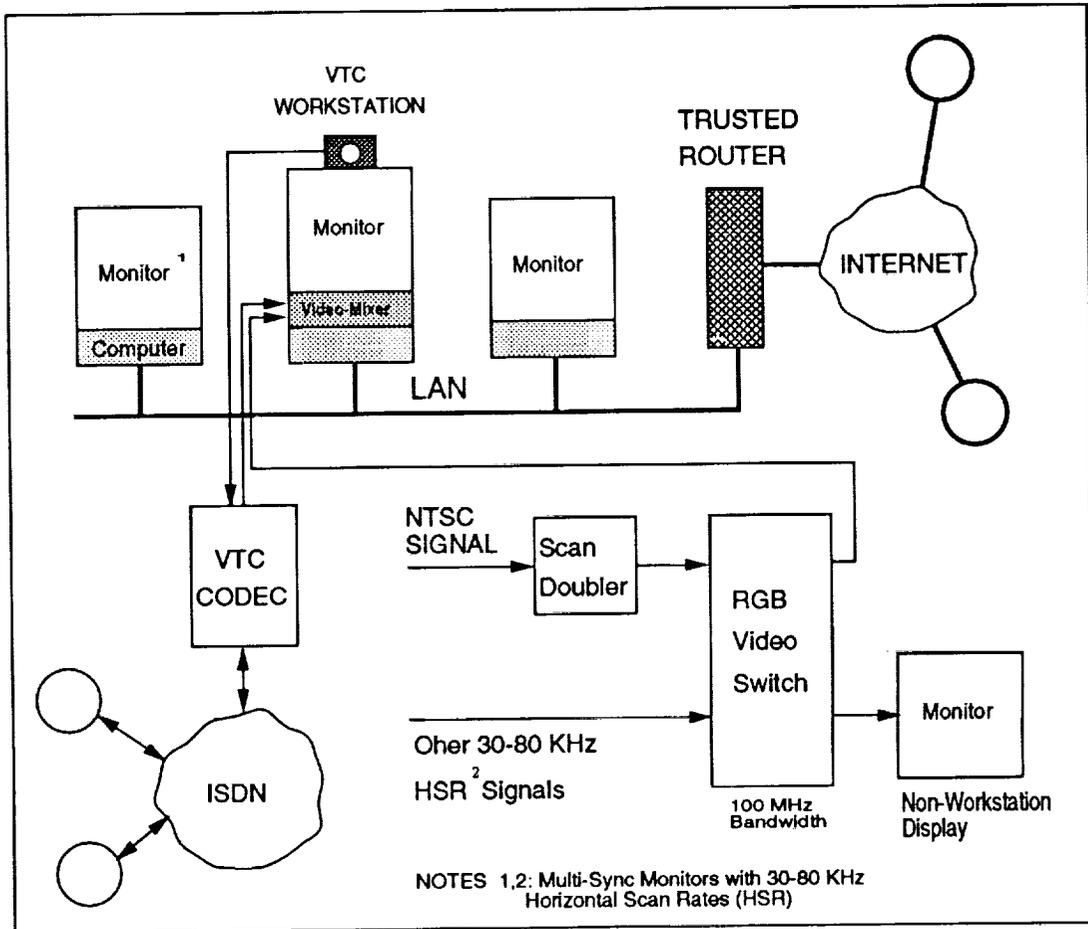


Figure 1: Typical system configuration.

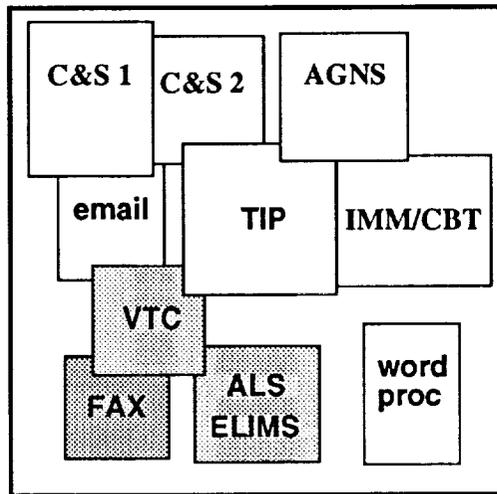


Figure 2: Depiction of some X Window services that could be made available to each workstation. Legend: C&S = Control & Status, email = electronic mail, VTC = Video Tele-Conferencing.

- Video Teleconferencing (VTC)
- Electronic Mail (email)
- File Transfer Protocol (ftp)
- Remote login (telnet)
- WYSIWYG Word Processing
- Weather Reports

An *operations* console should have all the above services plus access to

- Several Control and Status GUIs as dictated by the position responsibilities
- Facilities Automation System (FAS)
- Automated Ground Network System (AGNS)

TIP is a Mission Operations and Data Systems Directorate – Code 500 Level I Project. It was conceived as a comprehensive approach to unifying the technical information systems across Code 500. TIP is building an inter-organizational information infrastructure whose operations concept is based on four tenets: information capture at the point of origin, interoperability of tools, reusability of technical information products and electronic connectivity. TIP information infrastructure functions span the four domains of technical information processing: creation/revision, distribution, use and management. The reuse of contract data deliverables, especially technical information manuals, engineering drawings and their associated computer files, is a major goal of TIP and results in significant cost savings. In addition to creativity standards and specifications, TIP is providing integrated services such as text and graphics scanning, networked printers and plotters, engineering CAD symbols libraries, electronic publishing, a master index of documents and drawings, document number administration, and automated distribution.

The ALS modules will provide information regarding Customer Orders, Order Status, Asset Availability, Technical Information and hypertext Customer Service Handbook. A CD-ROM system is provided as a subscription service by Information Handling Services (IHS). Some of the products currently available are

1. IC/Discrete Parameter Database
2. Vendor Master Directory, full text OEM and Distributors Catalogs
3. NASA Documents
4. Department of Defense Standards and Specifications
5. GSA Source One
6. RECAL/Z (Resistors and Capacitors)

Both text and images are available for PC, Macintosh and UNIX platforms. Future plans call for the information to be delivered via LAN/WANs to authorized users.

E-LIMS will be the site-based portion of the Logistics Engineering Support System (LESS) that supports the information management needs of the site logistics activities. E-LIMS will maintain the Site Inventory Database (SIDB) and generate requisitions when predetermined resupply levels are reached (automatic resupply). E-LIMS will provide site personnel with the capability to query local inventory, issue items, place items into the inventory, and requisition items from the Logistics Support Depot (LSD). It will also enable the customer to query LSD-provided catalog systems (e.g., HAYSTACK or local CD-ROM database) as well as other stations' and LSD inventories. It will respond to inventory queries from other E-LIMS and the LSD. E-LIMS will be a PC-based client-server system of multiple, interconnected workstations at each site. It will provide a Graphical User Interface, mouse and incorporate automatic identification equipment (e.g., bar code) to facilitate the inventory control process. It runs under OS/2 and uses the DBM relational database management system. E-LIMS is currently under development and partial functionality exists. The first system will be installed at the White Sands Complex by the end of 1993.

IMM/CBT is provided by the Networks Training and Test Facility (NTTF). A prototype system running on a Mac Centris 650 equipped with a CD-ROM drive can provide audiovisual training ranging from introductory-level Networks system description (for example a TDRSS Orientation Course) to fully interactive testing, scoring and ranking on particular hardware such as the Command Data Formatter (CDF). A typical session shows scanned images of the system board-level components with whom the student can interact measuring signals, applying power, troubleshooting, etc. This approach can provide the required simulation based training leading to certification for mission critical systems which can not be made available for training.

ISDN-based *dial-up* Video Teleconferencing should be provided to selected positions such as station directors, network managers or mission controllers. VTC with its graphic and image capture capabilities is the obvious progression to the voice telephone and FAX. Additionally, NASA Select or any other RGB signal present at the location's TV switch could be selectively made available to authorized users as a *Picture-in-Picture* window. We propose to use INTERNET Electronic Mail which provides world-wide access to commercial, educational, scientific, military and government facilities. Its geographical coverage is much less restricted and it is far more friendly than our current NASAMAIL facilities. With INTERNET we also get fast file transfer service (ftp) and remote login capabilities.

Operation consoles can benefit from the above services while performing their required control and status functions. Two other systems, FAS and AGNS could physically decentralize while logically centralize Facilities and Operations.

FAS is composed of four subsystems including:

1. Computerized Maintenance Management System (CMMS)
2. Electronic Drawing Management System (EDMS)
3. Facilities Management Information System (FMIS)
4. Equipment Instrumentation System (EIS)

The CMMS will be used by station personnel to initiate and track work orders, manage preventive and predictive maintenance and for work planning and scheduling. The CMMS will electronically interface with the station logistics systems to determine the availability of parts required and for subsequent requisitions.

The EDMS will be used at GSFC, by support contractors and at each Network station. The EDMS will allow its users to review and plot selected copies of facility as-built drawings. The EDMS will also allow facility supervisors at each station

to annotate drawings in conjunction with a Facility Change Request (FCR). The NMOS support contractor will maintain a master set of as-built drawings and will make distribution of duplicate raster formatted copies (electronically) to GSFC and to each respective station as necessary.

The FMIS will be used by facility personnel at each Network station and by GSFC personnel to track special facility events, assist in facility project planning and handle all phases of FCR processing (EDMS will provide access to related drawings). The FMIS will also be used to perform budgetary analysis and forecasting and provide advanced reporting capabilities.

The EIS will be used at each Network station to monitor and control facility support equipment such as power distribution, air conditioning, antennas, fuel systems, station engine/generator systems, fire detection systems, etc. The EIS will also monitor environmental parameters (inside and outside) such as temperature, pressure, humidity and air quality. A significant feature of the EIS will be its ability to track and provide early problem detection of critical support equipment. This will permit advanced repair and correction (before failure) thereby avoiding a potential station down time.

The intent of the AGNS is to reduce station life cycle costs. Automated Monitoring and Control (M&C) is one of the key strategies proposed for achieving that goal. Other strategies include the use of Commercial Off-the-Shelf equipment in preference to custom-developed equipment, consistent application of standards to improve interoperability and the maintenance of configuration information in easily modified database tables.

AGNS uses commercial workstations (SUN Sparc) for operator interfaces, commercial control processors (VME computers) for real-time equipment interfaces and commercial M&C software. Development engineers define the behavior of all interfaces between the real-time computers and the equipment being controlled using

standard database tools (dBase IV) and draw display screens using software that mimics the behavior of typical computer drawing programs.

The workstations communicate with real-time computers using Ethernet and TCP/IP. Separating real-time processing responsibilities (done in the real-time computers) from the display functions (done at the workstations) permits us to place operators and their workstations at any location that can be reached via local or wide area network. Only the real-time computers need to be co-located with the equipment being monitored or controlled.

4 Security Issues

There is the need to provide National Resource Protection and Computer Security for systems connected to the INTERNET. The usual way of addressing this problem is by using trusted bridges and routers. A commercial system, the *FIREWALL* Computer[4] has been mentioned as a candidate to perform packet security filtering. This system provides some measure of isolation between the a LAN and INTERNET. The FIREWALL system controls and monitors all incoming and outgoing traffic but only allows connection of those hosts and services which are cleared to be interconnected. Another possibility is the Compartmented Mode Workstation (CMW)[5]. This System is rated as B1+. UNIX machines such as the SUN can be minimally configured as C2 systems by themselves without using any external devices[6].

5 Implementation Example

We recently demonstrated a prototype "Multimedia" system for the NASA Station Management Conference. The demonstration included a subset of the most likely services required in future NASA Control Center and Station administrative consoles. Some of the services shown were the Technical Information Program electronic

library, the Automated Logistics System showing scanned commercial catalogs directly from CD-ROM, an Interactive Multimedia Computer Based Training audiovisual application, ISDN based Video Teleconference and several other utilities such as live television in a window, a typical Control Center Graphical User's Interface simulating Space Network message flow, the Moving Picture Expert Group (MPEG) video-compression standard, FAX, Word-Processing utilities, and the latest weather including cloud cover and ground observed winds and temperatures. These utilities were complemented with email and electronic talk and were seamlessly integrated and concurrently displayed on a single screen using the X Window System.

The Local Area Network used for the demonstration contained a typical heterogeneous mixture of machines including 486 PCs, Macs and SUN SPARCstations aimed at showing the concept of system interoperability and compatibility. Each host with its appropriate TCP/IP driver was networked using *thinnet* (10₂) Ethernet. The PCs were running DOS 6.0 and Microsoft Windows Version 3.0 and the Macintoshes were running Mac OS 7.0. The SUNs were running SunOS Release 4.1.1. For the PCs we used DESQView/X 486 X-Server Software. This package also provides a limited DOS and Microsoft Windows X-client capabilities. For the Macintosh we used Planet-X client software and the Macintosh eXodus X-server.

6 Conclusion

Based on the very positive feedback and acceptance of the "Multimedia" workstation concept shown at the NASA Station Management Conference it is recommended that INTERNET and the X Window System be used by the NASA Networks Division to further solidify a distributed open computing environment among our stations.

Future availability of the LAN/WAN interconnection through the AGNS-provided INTERNET will integrate the Merritt Island and Bermuda Stations with the God-

standard Center Network Environment (CNE). There is also active interest in providing INTERNET service to other Space Network (SN) locations. We need to assess any security risks and weight the cost/benefit tradeoffs involved in making this decision prior to connecting these sites to the INTERNET. A local administrative network at an SN site does not necessarily imply an INTERNET connection, however the site could immediately benefit from resource sharing and vendor interoperability if a TCP/IP LAN solution is used.

The open nature of the X Window System with its high degree of vendor independence, modularity and expandability not only will improve the operations at NASA control centers and stations but will signify a radical departure from proprietary and custom architectures. As a net result, an implementation based on the X Window System could imply lower life cost cycles and a more reliable distributed computing environment. An X Window System implementation will also emphasize NASA's commitment to Total Quality Management.

References

- [1] *X Window System User's Guide*, V. Quercia and T. O'Reilly, O'Reilly & Associates, Inc. 1990
- [2] *Space Network Control Systems (SNCS) Human-Computer Interface (HCI) Assessment Report*, J. Freedman et.al., Stanford Telecom, August 14, 1992
- [3] *NASA Scientific and Engineering Workstation Procurement*, Contracts NAS5-37001 through NAS5-37009
- [4] *Local Network Security Considerations*, D. McNulty et. al., Stanford Telecom, November 6, 1992
- [5] *Compartmented Mode Workstation*, D. McNulty et. al., Stanford Telecom, May 7, 1993
- [6] *System and Network Administration*, SUN Microsystems, Revision A, March, 1990

THE AUTOMATED GROUND NETWORK SYSTEM

Miles T. Smith

Telecommunication Systems Branch
Goddard Space Flight Center

Peter N. Militch

Engineering Section
AlliedSignal Technical Services Corporation

Summary

The primary goal of the Automated Ground Network System (AGNS) project is to reduce Ground Network (GN) station life-cycle costs. To accomplish this goal, the AGNS project will employ an object-oriented approach to develop a new infrastructure that will permit continuous application of new technologies and methodologies to the Ground Network's class of problems. The AGNS project is a Total Quality (TQ) project. Through use of an open collaborative development environment, developers and users will have equal input into the end-to-end design and development process. This will permit direct user input and feedback, and will enable rapid prototyping for requirements clarification. This paper describes the AGNS objectives, operations concept and proposed design.

1 Introduction

The National Aeronautics and Space Administration (NASA) Ground Network has gradually evolved over the years to satisfy changing requirements and to accommodate new technologies. It has effectively satisfied mission requirements and has generated an enviable record of station availability and performance. However, the evolutionary process has resulted in a diverse collection of aging, custom equipment that is becoming increasingly more difficult and expensive to maintain. Technology advances over the past few decades allow for "better, faster, and cheaper" ground tracking stations. The goal of the Automated Ground Network System (AGNS) project is to use these advances to implement dramatic improvements in station life-cycle operating costs and efficiencies. Since technology will continue to advance, and requirements will continue to change, the project must provide for continuous technology insertion with minimal disruption to routine operations.

Every automation project is unique because every system has unique requirements and constraints. Although other automated tracking systems exist, such as the United States Air Force's Automated Remote Tracking Station (ARTS) and the Wallops Transportable Orbital Tracking System (TOTS),

these systems are not designed to accommodate the Ground Network's unique requirements. The Ground Network tracking stations at Merritt Island and Bermuda are unique because they support launch and landing of manned spacecraft, as well as provide orbital support for a very wide variety of spacecraft. By contrast, the ARTS was designed to provide repetitive orbital support for many different spacecraft with similar or identical receive and command formats. The TOTS was designed to provide repetitive orbital support for a few spacecraft, and it was optimized for portability and simplicity.

During the history of the Ground Network, there have been numerous attempts to automate portions of the stations and their operations work-loads. Station requirements have always been fluid, and the technology did not always exist to cost-effectively automate work to the extent desired. The AGNS project has paid close attention to the lessons of history. The most important of these is that the stations must have significant input in the early stages of the project. Much of the work at each site is repetitive, but there are day-to-day variations. A design that is inflexible, or inappropriate, will not achieve the desired efficiencies and may not work at all. Significant and continuous operator input is in keeping with the project's TQ approach, and permits NASA to draw upon hundreds of person-years of experience available at the sites. This commitment to learning from the users and implementing tools to ensure their input is critical to project success.

This paper describes the AGNS project's objectives, its scope, and its approach to reducing life cycle costs. Section 2 describes work at the current Ground Network stations that is amenable to automation, and it outlines AGNS project objectives as they relate to this work. Section 3 describes the scope of the AGNS project and the major subsystems the project will implement. Sections 4 through 6 describe the AGNS Development Facility, Maintenance and Support Network and the Monitor and Control Subsystem.

2 Project Objectives

The AGNS project, managed by the Telecommunication Systems Branch (Code 531) at GSFC, will enable more cost effective GN tracking station support. The objectives of the AGNS project are to:

- a. Significantly reduce station life-cycle costs.
- b. Improve station reliability, maintainability, and availability.
- c. Enable and enforce the use of internationally-recognized communications standards and protocols.
- d. Provide a flexible station architecture that can readily accommodate future requirements

and technology with minimum effort and cost.

Operations and maintenance costs for the GN tracking stations at Merritt Island and Bermuda and the wing site at Ponce De Leon are driven by many factors. These include a tightly coupled and complex software architecture; the use of large numbers of different, custom communications interfaces; and labor costs that are driven by the peak work loads encountered during preparation for, and support of, Shuttle launches and landings.

First, the existing stations' architecture is tightly coupled. Changes in one subsystem often impact equipment and procedures elsewhere on the site. This necessitates extensive regression testing when making software modifications. Such testing often impacts external users as well as station operations.

Second, since existing station equipment uses a variety of custom communications interfaces and protocols rather than interfaces based on internationally-recognized standards and protocols, it is difficult to use commercial off-the-shelf (COTS) equipment for system upgrades. Some degree of tailoring is invariably necessary. This increases development effort, complexity and cost. A further consequence of these low levels of interoperability is that it is very difficult to reallocate processing among subsystems, so they are often replaced on a "one-for-one" basis, perpetuating the problem.

Finally, labor costs are the single largest component of station operating expenses, and since the number of operators is determined by peak, not average work loads, the cost to operate the GN is significant.

The AGNS will simplify station maintenance and reduce operations costs. The project will:

- a. Use loosely coupled subsystem building blocks to simplify new development and minimize requirements for regression testing.
- b. Encapsulate station functionality in "black boxes" that hide implementation details from other subsystems and provide well defined messaging interfaces for exchange of data.
- c. Automate station configuration, testing, fault isolation and recovery.
- d. Utilize expert systems to capture operator knowledge, filter display information to avoid operator overload and to respond to contingencies.
- e. Enforce the use of common, commercial standards, interfaces and equipment.
- f. Provide centralized monitor and distributed control capabilities.

3 Project Scope

The AGNS project addresses only the GN tracking stations at Merritt Island and Bermuda. It is not within the scope of this project to modify configurations, interfaces and operational procedures at other GN sites or other locations such as Johnson Space Center (JSC), Goddard Space Flight Center (GSFC), Kennedy Space Center (KSC), Department of Defense (DoD) sites or in the NASA Communications (Nascom) network. The AGNS project provides capability for external operators (at the Network Control Center for example) to access information and control activities, if required.

The AGNS project presently consists of three subprojects: the AGNS Development Facility (ADF), the AGNS Maintenance and Support Network (MSN), and the AGNS Monitor and Control Subsystem (MCS). The ADF will provide developers and the sites with the tools necessary to prototype, develop, test, and maintain other subprojects in support of the system. The MSN will permit station personnel to share these tools, enabling their early and continuous participation in system requirements analysis and design. The MSN will also enable sustaining engineering personnel at contractor facilities to remotely troubleshoot and monitor station subsystems. The MCS will provide interactive, real-time monitor and control of stations' equipment.

4 AGNS Development Facility

The ADF is a distributed facility with components at GSFC, Merritt Island, Ponce De Leon and Bermuda. The ADF at GSFC will be a remote extension of the stations' monitor and control subsystems. Developers will work interactively with site operators and engineers to prototype, evaluate, test, iterate and refine:

- a. System displays and control procedures.
- b. Expert system rule bases for operations resource scheduling and fault management.

The ADF equipment is all COTS and will be database-driven. Resource editors and commercial databases will define and store the parameters that define system appearance and behavior. This will permit rapid changes, but will require only minimal regression testing. This concept has already been successfully demonstrated in development and implementation of the Gamma Ray Observatory Remote Terminal Subsystem in Canberra, Australia. Development cycle times were reduced from weeks to only hours through the use of database-driven subsystems, electronic interaction with site equipment from GSFC and electronic exchange of system development and project management information.

5 AGNS Maintenance and Support Network

The AGNS Maintenance and Support Network (MSN), as shown in Figure 1, will consist of a number of workstations and Ethernet local area networks (LANs) interfaced to the National Science Foundation Internet via a Program Support Communications Network (PSCN) Gateway.

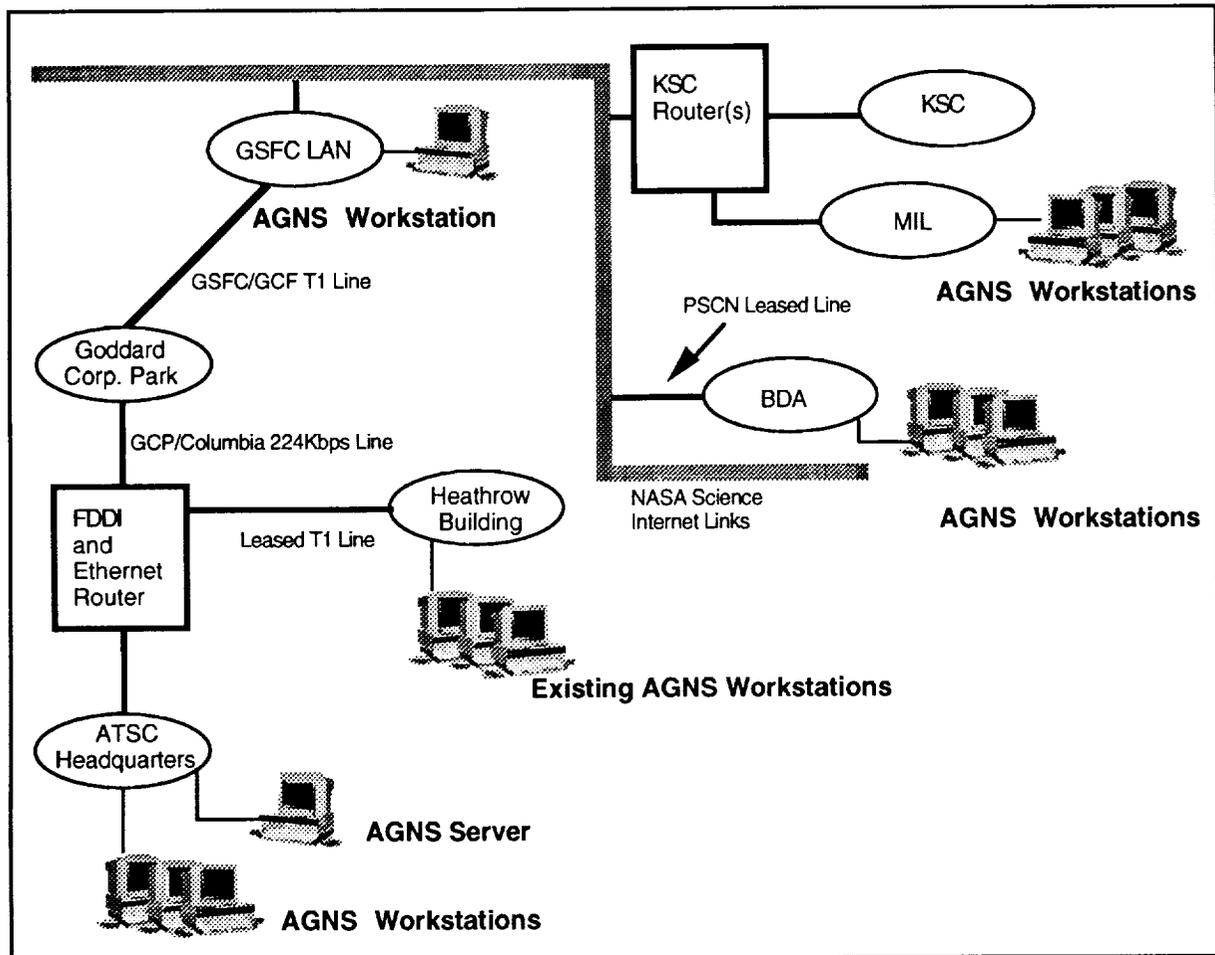


Figure 1 AGNS Maintenance and Support Network

The MSN will reduce life cycle costs by:

- a. Providing effective electronic connectivity for electronic mail services and electronic conferencing, thereby improving communications between the sites and GSFC.
- b. Providing seamless integration with major Networks Division (Code 530) Management Information Systems including the Technical Information Program (TIP), the Engineering Change Automation System (ECAS), the Facilities Automation System (FAS), and the

Enhanced Logistics Information Management System (ELIMS).

- c. Reducing development cycle times
- d. Enabling remote troubleshooting and reducing response times when problems occur.
- e. Improving system performance analysis.
- f. Sharing project management information between GSFC, contractors and the sites.

For security reasons, the AGNS Maintenance and Support Network will not be connected to the AGNS Monitor and Control Subsystem network until a trusted interface is identified or developed.

6 AGNS Monitor and Control Subsystem

The AGNS MCS approach is based on careful review of current station operations. During project planning activities, site personnel and developers attempted to define the capabilities of successful operations teams and important features of their approach to identify key attributes of a good MCS design.

The lessons of two decades of station operations are clear. To significantly reduce operations staffing, any MCS must provide complete equipment control. Feedback must be comprehensive and fast, but must not overload operators with status information. Prioritizing and hardwiring status information works only until an unanticipated problem occurs. All the status information must be available for selection, but only that data desired at any specific instant should be presented to the operator. Similarly, the control functions needed at any moment should be the easiest ones to access.

It is valuable to examine why these lessons have not been applied in the existing GN. The key reasons appear to be:

- a. Limited processing resources. Computers built in the 1970's did not have the speed or memory required to sample data and status, make preliminary analyses, draw conclusions and offer suggestions to operators.
- b. Primitive development approaches. No development methodologies existed to rapidly, safely, and cheaply iterate monitor and control approaches. The developers' best guesses by Critical Design Review time defined the subsystems and their operating modes for the remaining life of the equipment.

The proposed monitor and control approach for the AGNS is to provide all of the relevant equipment audio and visual cues that exist today to any operator, independent of location. Sites

will specify different kinds of support scenarios and will identify information that is important in those contexts. Since it is impractical to define all possible scenarios during the design phase, all screen displays and control interactions will be defined in operator-accessible databases. Users will define all the interfaces through an iterative, prototyping approach. The equipment and software configuration and behavior will be defined by parameters stored in external databases. By modifying information in the databases, site personnel will be able to modify station processing and displays. This will permit them to make changes locally in minutes or hours. Presently, it can take weeks or even months to modify and test system software. New configuration management procedures have been defined and will be used to manage this shortened development cycle.

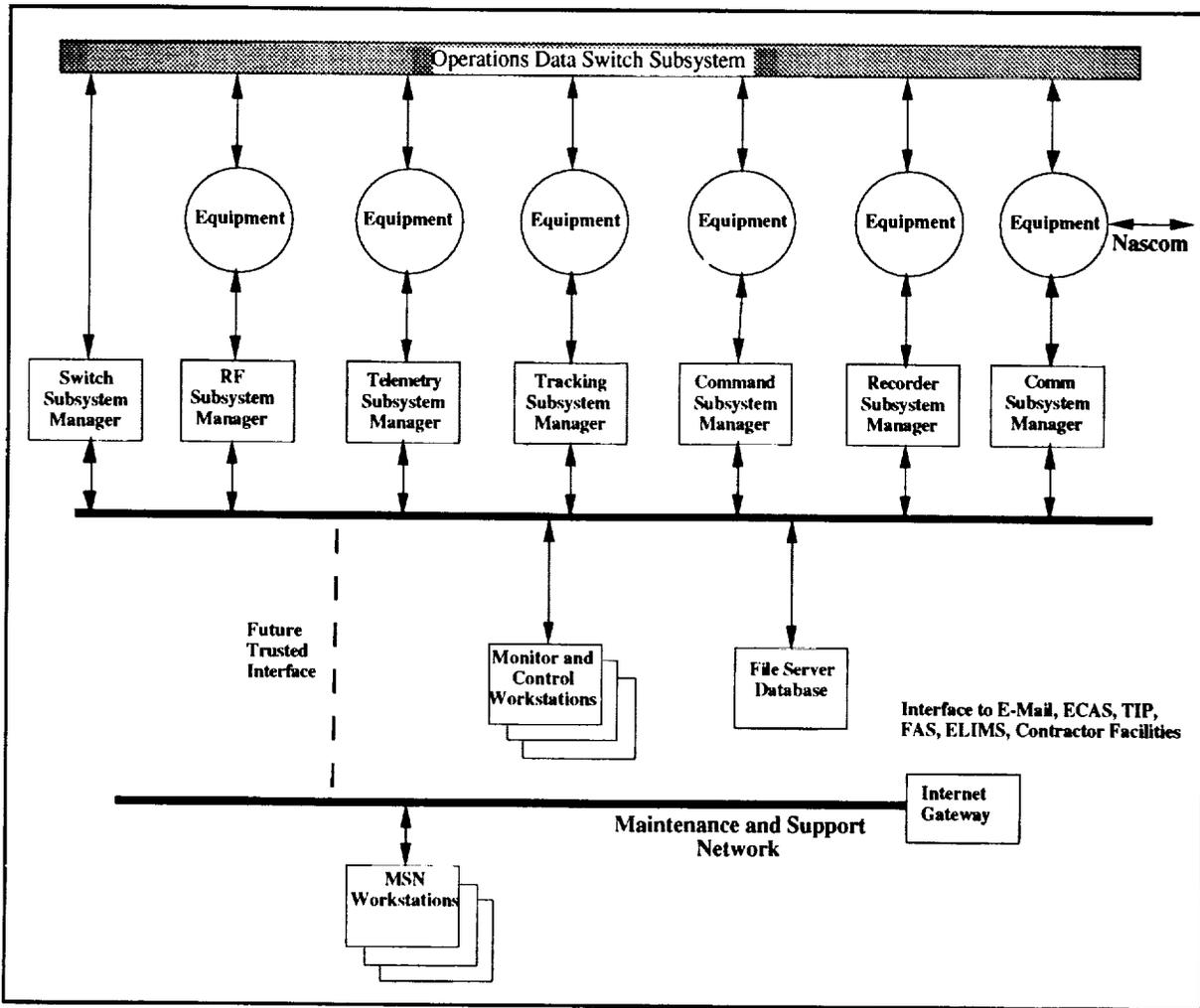


Figure 2 AGNS Station Architecture

As shown in Figure 2, the AGNS MCS will comprise a file server/database, a series of distributed subsystem managers, a small number of centralized monitor and control workstations, and an

Operations Data Switch Subsystem – all interconnected via a communications network using standard communications protocols. The database will store important subsystem equipment configuration parameters for all supported spacecraft, as well as the station's mission support schedule. The subsystem managers will encapsulate the details of the individual subsystem components, will provide well-defined monitor and control interfaces, and will utilize expert systems to help automate subsystem configuration, testing, fault detection and recovery.

The subsystem managers will monitor and control their subsystem components via real-time databases. The real-time databases will function as a "presentation layer" between operator workstations and equipment. They will provide status information to the centralized monitor and control workstations, which will display the status using a standard graphical operator interface. These workstations will also transmit control information, via the real-time databases, to the subsystem managers.

The system schedule will define overall operations requirements. The subsystems will decide locally how to implement those portions of the schedule that they can accommodate. During equipment configuration and fault recovery operations, subsystem managers will transmit status messages to the Switch Subsystem Manager. This manager will interconnect the subsystem processing strings as required, using the Operations Data Switch Subsystem. A detailed record of all configurations will be maintained for subsequent fault or performance analysis, should this be necessary.

This object-oriented architecture will reduce coupling between subsystems and will create a hierarchical monitor and control design structure. In a typical operations scenario, the schedule, which is a high-level document, will be received at the AGNS station from the Network Control Center (NCC). This schedule will be stored in a site database where it will be accessible to each of the subsystem managers. Each subsystem manager will query the database using a Structured Query Language (SQL) protocol to obtain relevant portions of the schedule. Each manager will know, from its local knowledge base, the necessary frequencies, bit rates, and other configuration parameters required for its subsystem to satisfy its specified support functions or requirements. The managers will then allocate resources to each function.

Subsystem managers will be aware of the status and allocation of all components they control, and they will be locally responsible for resource allocation. Status information will be available to external subsystems if desired, but this information will have to be requested from the manager.

At the requested time of support, each subsystem manager will read its schedule requirements from the central database, will allocate and test subsystem components, and will send a status message to the Switch Subsystem Manager informing it of which subsystem components it is contributing

to the overall processing string. The Switch Subsystem Manager will use this information to control the actual data paths in the Operations Data Switch Subsystem.

7 Conclusion

The AGNS project will implement a modular, expandable, flexible station architecture. It will employ expert system technologies to capture operations knowledge and reduce personnel peak work loads. The project will implement a collaborative development environment to shorten development cycles and converge on effective engineering solutions in the shortest possible time. The architecture will permit continuous technology insertion, ensuring that NASA can make maximum use of new, cost-effective technologies as they become commercially available.

Although the AGNS project will provide the Ground Network with a new infrastructure that will enable significant life-cycle cost reductions, these cost reductions will not be fully realized until the old, custom equipment is replaced with new, standard equipment. The Telecommunication Systems Branch plans to replace this equipment over the next few years. A "better, faster, and cheaper" Ground Network should soon be a reality.



Mission Operations Support Area (MOSA) for Ground
Network Support

Robert D. Woods and Susan A. Moser, AlliedSignal Technical
Services Corporation

The Mission Operations Support Area has been designed utilizing numerous commercial off the shelf items (see Figure 1), when possible, providing ease of maintenance and upgrade-ability. At its inception, all equipment was at the fore-front of technology. The system was created to provide the operator with a 'State of the Art' replacement for equipment that was becoming antiquated and virtually impossible to repair with new parts because of unavailability. Although the Mini-NOCC provided adequate support to the Network for a number of years, it was quickly becoming ineffectual for higher data rate and non-standard missions. The MOSA will prove to be invaluable in the future as more and more missions require Ground Network support.

For the past several years, NASA has provided Operational and Technical support to its Ground stations utilizing the Mini-NOCC telemetry, command, tracking, range safety, and log/delog data systems shown in Figure 2. Specific functions include, station sub-system verification, system verification, Mission support verification, personnel proficiency training, mission support validation, and pre-mission, mission, and post mission fault isolation and analysis. These functions are provided for Space Shuttle, Expendable Launch Vehicles (ELV), and their payloads as well as station changes and upgrades.

Early in 1990 the Small Explorer project requested the assistance of the Networks Division in providing these support functions for a series of ground supported high data rate CCSDS compatible payloads. With the current Mini-NOCC being restricted to 224kb data streams and unable to process CCSDS recommendations, it was evident that not only a more enhanced data system would need to be developed to meet these requirements, but this new system must be designed and

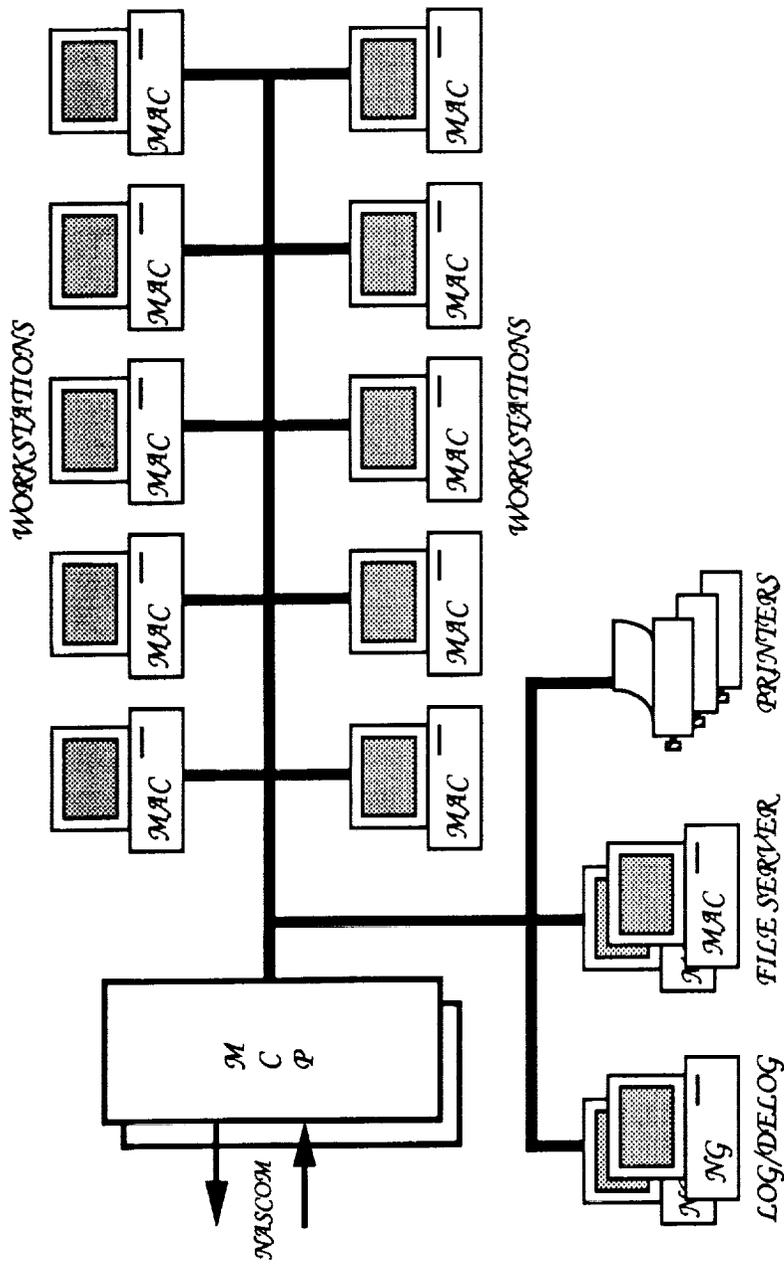


Figure 1. MOSA Basic Plan

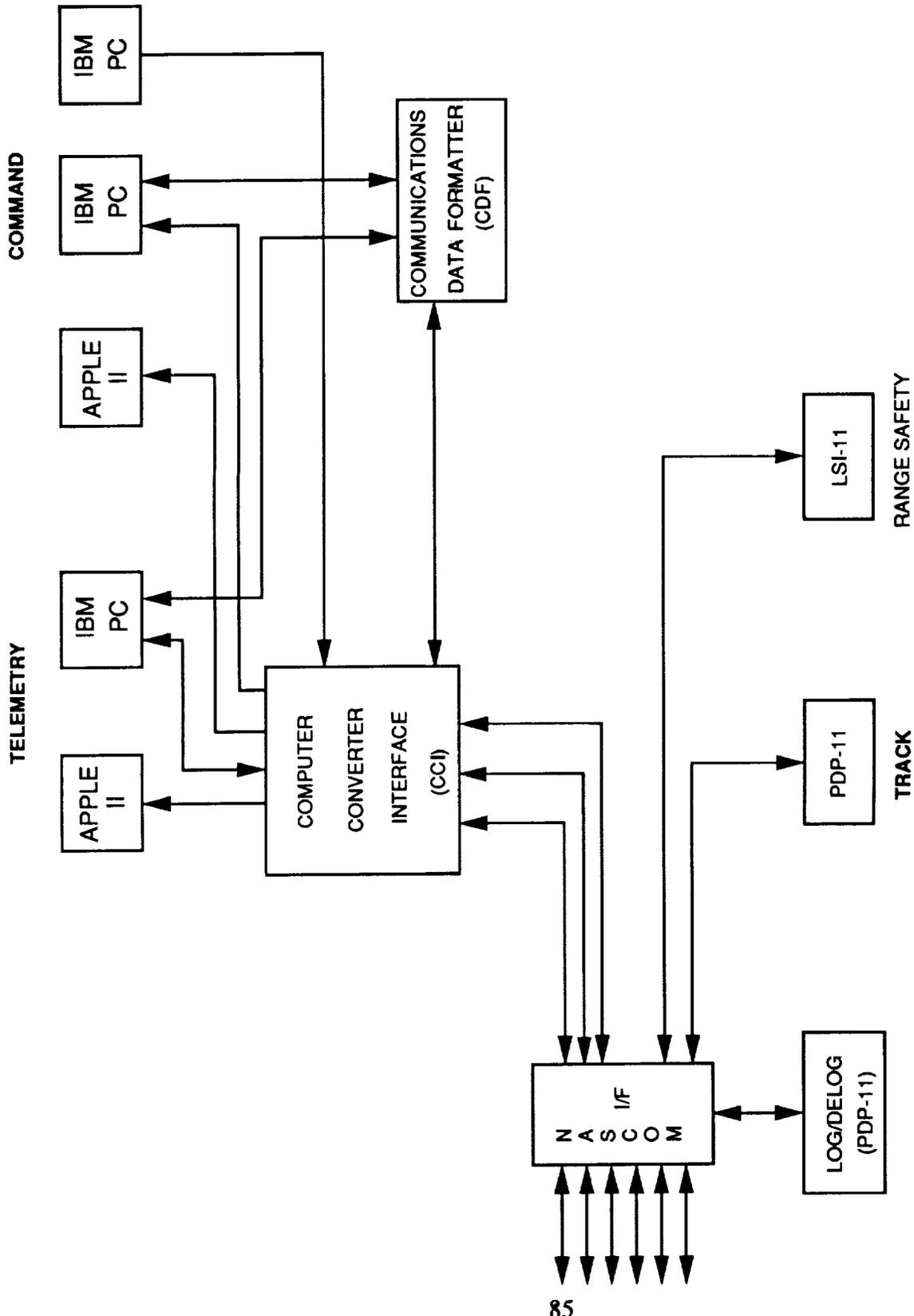


Figure 2. Mini-NOCC Configuration

implemented in a relatively short period of time in order to meet the established SMEX mission manifest. With a challenge of this magnitude, it was decided that a team of qualified engineers from both NASA and AlliedSignal Technical Services (ATSC) be formed to tackle the project. Thus came in to being, the Mission Operations Support Area (MOSA).

After investigating all avenues, a final design was agreed upon. The design, as reflected in Figure 3, consisted of multiple workstations tied together via an Ethernet interface to an intelligent Front End Processor controlling all incoming and outgoing data. Each workstation would consist of a Macintosh or PC based system with dedicated video monitors. Ten identical Macintosh workstations would be utilized as the basic support system providing telemetry, command, and track functions. The log/delog system would be comprised of two Northgate 486 PC's and hard drives capable of logging 1GBytes of data. In addition to these workstations, it was decided to incorporate another to act as a system file server that could be used as a massive system data base providing multiple system and program aides. To accommodate the system, new consoles had to be designed and fabricated. And to further assist in meeting the projected mission timeline, the decision was made to relocate the MOSA instead of interfering with ongoing mission support in the Mini-NOCC.

The following paragraphs describe in high level detail individual MOSA sub-systems and their relationship to each other and the overall system concept.

MOSA COMMUNICATIONS PROCESSOR

The MOSA Communications Processor (MCP) provides the interface to all external Nascom circuits. Two redundant MCP's were developed around a Motorola 32MHz MC68030

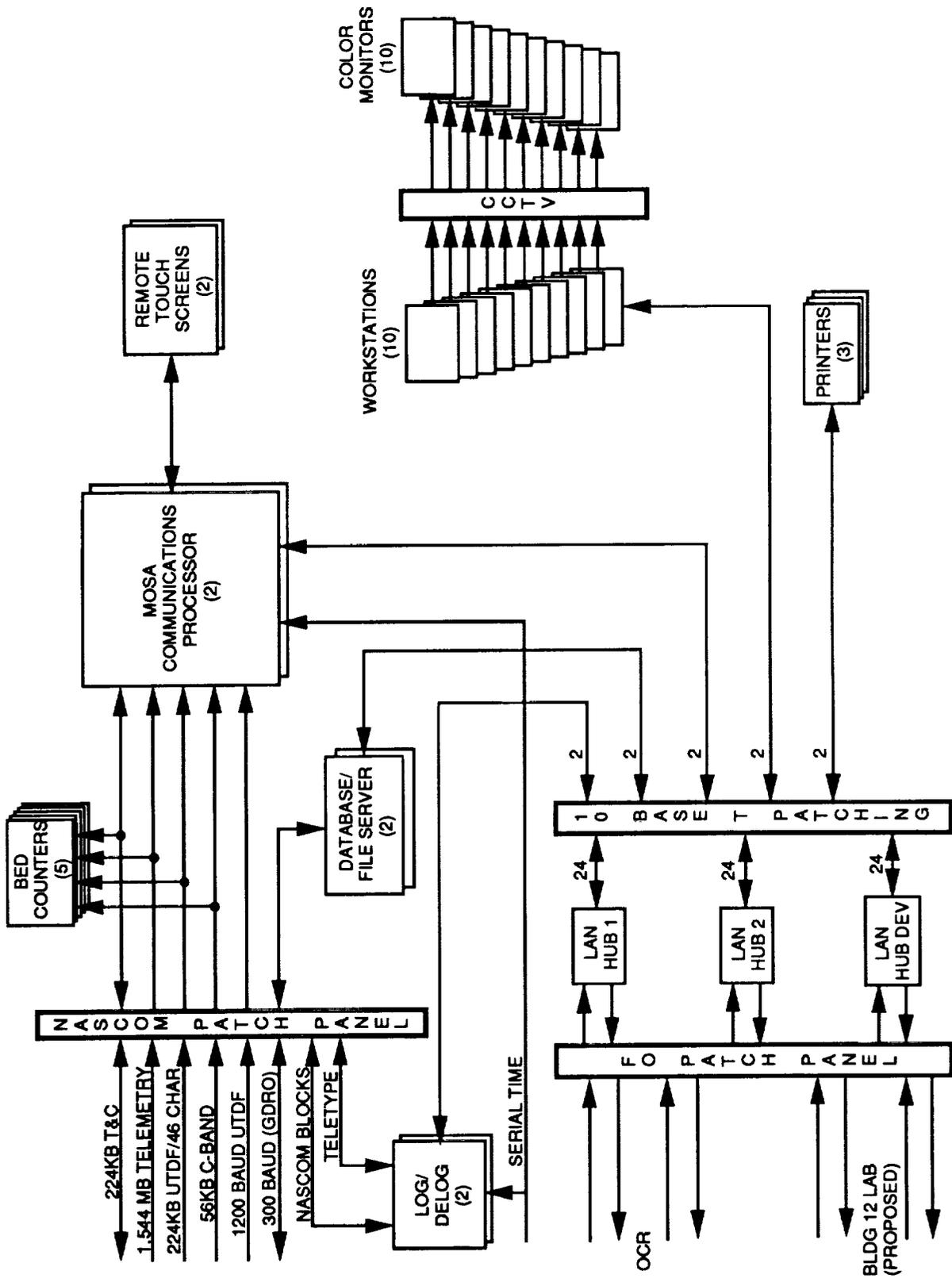


Figure 3. MOSA Detailed Block Diagram

microprocessor with 16 Mbytes of RAM for subsystem communication. Of the two processors within each MCP, reference Figure 4, one acts as the master controller while the other is the slave. The master processor controls all communications to the workstations, maintains the subsystem data requests, and controls the status and diagnostic displays at each console. The slave processor analyzes the data as it is received, builds the frames of data for display, receives the asynchronous tracking data, and coordinates diagnostic testing. The MCP stores the analyzed data internally and provides it to the workstation upon request.

The operator control of the MCP is provided by touch screens. The touch screens are composed of plasma displays with infrared sensors located around the perimeter of the screen for sensing the menu selection made by the operator. The touch screens are located not only on the front of the MCP but also remotely from each console position. All touch screen functions are available, regardless of which display the operator utilizes. The MCP initializes the touch screen with the MCP identification number, the software version, the current GMT, provided by the internal Bancomm timing decoders, and the elapsed time from MCP activation. The MCP identifies which Macintosh workstations are currently configured to the MCP along with an incrementing counter for network messages transmitted to and received from the workstation. The same display will provide the time at which the Macintosh activated the connection and the number of windows active at that position. The system errors are displayed at the bottom of the main screen but can be expanded to thirteen lines with scrolling capabilities. The errors are identified and tagged with corresponding block time. The diagnostic display provides an analysis of the Intelligent Transmit/Receive boards. The test is configured

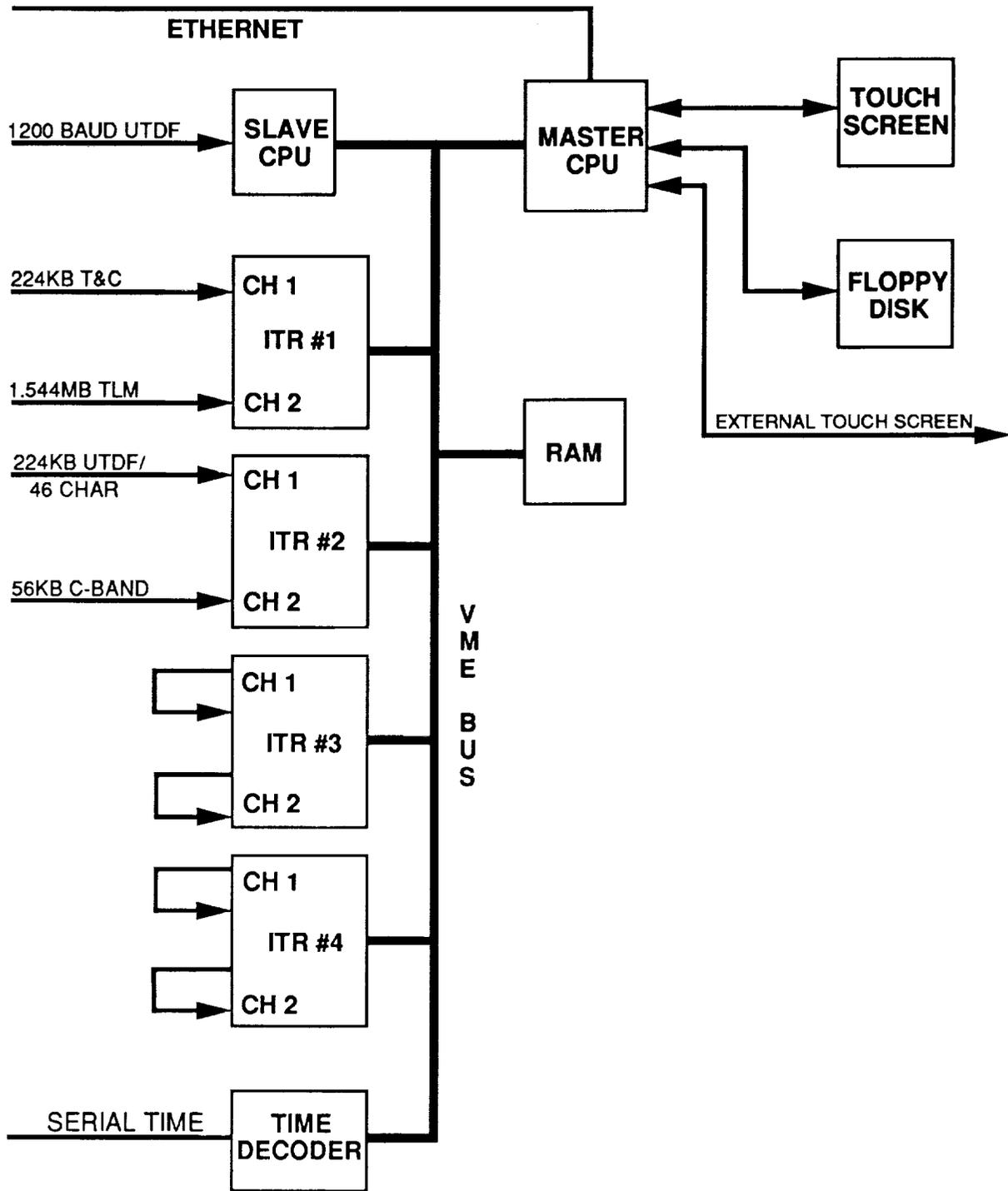


Figure 4. MCP Configuration

with a specific transmit port, receive port, clock rate, and receiver time out. The system performs a bit error rate check on the ports and delivers the results to the operator for analysis. The data connection display identifies which of the Input/Output (I/O) ports are active according to the stream name and/or clock activity. In addition to counting the number of elements received and transmitted, the MCP counts the number of Cyclic Redundancy Check (CRC) errors received on each channel. The system is capable of resynchronizing a stream of telemetry data in any standard format as well as CCSDS data up to 1.544 Mb/second. The MCP is capable of handling up to eight full duplex Nascom lines.

Within the MCP, Intelligent Transmit/Receive (ITR) boards are installed to provide an interface to Nascom equipment. These boards, designed and developed by NASA, communicate channel activity to the MCP's and to the workstations. The configuration is demonstrated in Figure 5. Each board provides full duplex synchronous channels for interfacing to NASCOM and resynchronizing telemetry. The board is a Harris RTX2000 micro controller programmed in Forth assembly language. Each transmit board can handle synchronous blocks up to 65536 bytes in length. This makes the boards practical for both Nascom blocked data and telemetry frames. CRC polynomials are generated by the ITR and inserted at the end of the Nascom blocks and telemetry frames. The transmit (or receive) clock can be externally supplied, internally created by the frequency synthesizer, or from a selected clock on another transmitter. The ITR receive capability is also 65536 bytes in length. Two 64-bit digital correlators allow detection of the selected synchronization pattern with programmable data and mask patterns and an allowable number of errors. The correlators operate in parallel to allow the detection of both true and inverted frame sync patterns. The receive channel on the ITR will also detect and correct for a

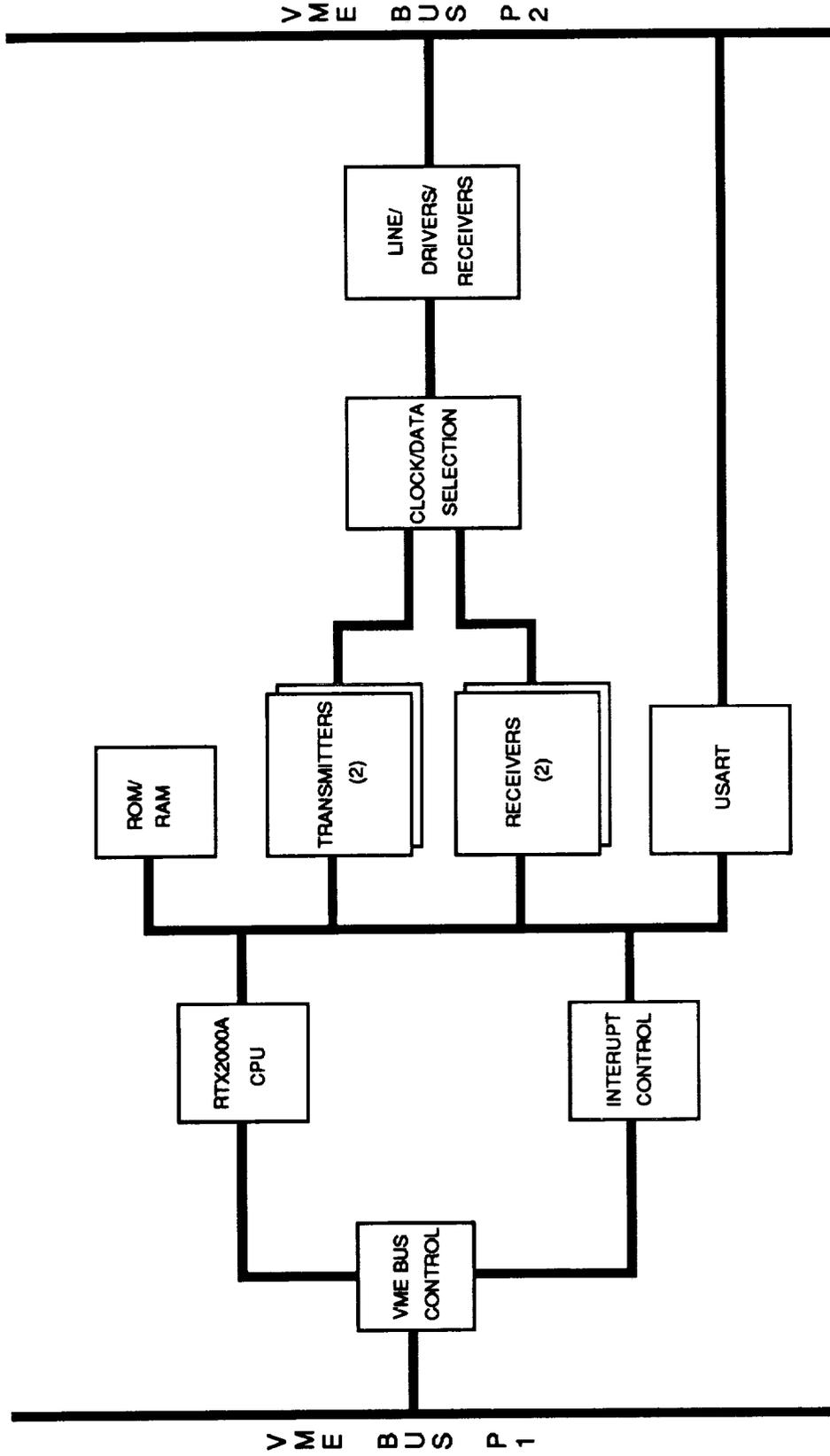


Figure 5. ITR Block Diagram

bit slippage of up to one bit in either direction. Receive CRC or Nascom polynomial errors are detected and reported to the MCP.

MOSA WORKSTATIONS

The Apple Quadra workstations replace the Apple IIe and IBM computers in the Mini-NOCC. The workstation is the primary operator interface in the MOSA data system. The operating system is Apple Computer's System 7. The MOSA software developed by NASA is the processing software that allows the user to interact with the database/server, log/delog, and MCP. The software was developed utilizing MacApp and C++ along with the importation of applications previously developed for existing station equipment. Upon launching the MOSA icon, an MCP connection must be chosen. If the operator later wishes to connect to the other MCP, the MOSA application must be ended and restarted. No connection is an option if the user is creating and storing data monitoring configurations for future use.

The MOSA software is a combination of all the functions previously supported by the Mini-NOCC (with the exception of Air-to-Ground). The operations are sorted into Block, Telemetry, Shuttle, CCSDS, Acquisition/Track, Range Safety, Command and general status information. Whenever a processing window is selected from the menus, the operator is prompted for the stream configuration information. The amount of parameter information requested depends upon the level of analysis.

The Block function processes Nascom 4800 bit blocks. A block show will display all blocks which meet the configuration criteria starting with the Nascom sync pattern. The scroll bars enable the user to examine the block in its entirety.

The block statistics window will process the indicated Nascom blocks and provide summary information. The information includes the number of blocks accepted, PEP errors, sequence errors, missed blocks, delta time errors, and Nascom frame sync bit errors. Counters in any statistic window can be reset individually or globally.

The Telemetry operation includes a frame show which displays the frame time and the raw frame starting at the frame sync. The telemetry frame statistics window processes the selected stream and indicates lock status, inversion, frames expected, frames accepted, true/inverted syncs, sync dropouts and frame sync errors. These two windows are specifically for throughput telemetry.

To support the shuttle program, several shuttle specific displays are required. An Operational Downlink (OD) show window displays the telemetry with subframe information. The Space Shuttle Main Engine (SSME) analysis tracks the number of valid frames and errors received for each engine along with a display of the frame its self. A statistical display of the data received from the stations gives the operator a quick view of the station performance. On another set of displays, the command, telemetry, and track site status messages (SSM's) are translated in alphanumeric values and displayed in full by scrolling through the window.

The CCSDS functions include both telemetry and command processing. The telemetry frame show and telemetry frame statistics windows are comparable to the throughput telemetry windows described previously. The CCSDS telemetry frame statistics window additionally includes counters for the number of frames and packets received relative to their virtual channel identifier. The telemetry frame header, telemetry packet header, command frame header, and command

packet header windows translate the header information into alphanumeric information based upon the corresponding CCSDS standard. The telemetry frame show, telemetry packet show, command frame show, and command packet show display the selected segment (frame and packet) of raw data for the given stream. The telemetry packet statistics displays the block, frame, and packet times along with the quantity of each application ID received in the telemetry stream. Once enabled, the lock/search history will annotate the actual GMT times for lock and loss of lock on a designated stream. The Command Link Transmission Unit (CLTU) statistics evaluates the telemetry stream. The CLTU portion of the stream is translated while the number of blocks received and the number of invalid CLTU's are counted. The Command Link Control Word (CLCW) display provides the parameters in alphanumeric symbols. The scrolling portion of the display contains the CLCW in raw format with GMT of receipt. Finally, the command history window contains a scrolling area for listing the command application ID, the commanded function ID, and the block time of the command.

The Acquisition/Track function processes all incoming tracking data for selection by the operator. The track status window indicates which tracking data types are being received and from which station the data originated. Valid data is written in green. A stream that has dropped out or has been interrupted will go to red for two minutes, then the stream will drop from the display list. An invalid LTAS stream will be displayed in black with only the valid bits displayed. Clicking on one of these listed streams will activate the tracking summary window. The track or Launch Trajectory Acquisition System (LTAS) summary window will translate the data selected into detailed alphanumeric displays. The summary windows can also be accessed directly from the Acq/Trk menu. If the window is selected from the

menu, the operator will be requested to choose the data for processing from a list of incoming streams. The options window allows the operator to select the parameter units for range, Doppler, and the angle coordinates. The track comparison window prompts the operator to select an acquisition message with which to compare the incoming tracking data. The message is computed and derived for the relative time found in the tracking data and then parameters are compared to the incoming station data. The transmit acquisition message window allows the selection of an acquisition message from the server database. The operator then selects which destination router is necessary for this message. The transmit button enables the sending routine.

Range safety processing is a detailed analysis of the 2.4 Kb streams utilized during ELV and Shuttle supports. The data is evaluated and displayed in alphanumeric format to enable rapid realtime analysis. The parameters are specified with the appropriate units. This function also has the capability to display the analog parameters within the stream as digital Cathode Ray Tube (CRT) strip chart signals. This process was originally performed with physical analog strip chart equipment provided specifically for this task.

Scientific spacecraft commanding allows the operator to transmit test commands to the stations and POCC's for verification. The commands are designed and created by the operator to match the POCC command structure. The transmit window will also confirm the validity of the station's command echo block. The Shuttle command function provides unencrypted modulation to the station and locks on the station's command echo. The modulation can be clocked internally or externally. A transmit window sends a command through the modulation for verification in the command echo portion of the window.

Further functions adapted from other systems in existence include the error history and the watch panels. The error history tracks the processing errors reported by the MOSA software and the MCP along with the time of their occurrence. The watch panels allow the operator to select a specific bit or group of bits to monitor within the block, frame, or packet. This utility greatly reduces the time spent manually delogging data or trying to see bit values on a constantly updating show window.

Unique capabilities are created under the MOSA software application. Since the capabilities are useful to all windows and operations, they are grouped under File and Tool menus. The MOSA software allows the user to configure a window and save the file under a descriptive name for use during operational support. When one opens a saved file, the software requests a confirmation of the configuration and then activates the window without additional entries. The print show operation will print to the Apple Laser writer the entire block or frame which is currently displayed in the active show window. The configure function supplies the configuration menu for the selected window in order to update the parameters for processing. Freeze/thaw will pause/restart the processing of the active window. The refresh rate or how often a window is updated can be increased and decreased, although this function is also affected by the number of windows open on the workstation. Specific warning messages are provided by the MCP to the workstation to alert the operator when changes are made to the MCP that will effect the workstation's current operations. A function is available to disable the warning windows but an audible alarm is still provided. Finally, the MCP can be configured to allow the processor to lock on a frame sync with a few incorrect bits. This effects all block synchronization that

the MCP does and therefore, effects all other workstations configured to this processor.

MOSA LOG/DELOG

The MOSA Log/Delogs element replaces the PDP-11/24 system originally used in the Mini-NOCC. Two Northgate 486DX 33MHz computers have the capability of logging Nascom blocked data and low speed teletype traffic. Each system contains two Nascom Interface Boards (NIB) and a timing board. The programs are created in a DOS environment with the Borland "C" compiler. The logger requires the use of MX, or Multitasking Executive, which was developed initially for the Telemetry and Communications Data System located at the NASA tracking stations in Bermuda and Merritt Island, Florida. This multitasking environment will allow the unit to log incoming data and also playback data, in separate operations, for the user. The 1 Gbyte hard disk drive is segmented for data storage. The high speed active area is used to log and store Nascom blocked data. This is the largest area since it is also designed to meet the requirement to log at least five minutes of 1.544 Mbyte data. The low speed active area is used to store the teletype tracking data. The last portion of the disk is set aside for the archive areas. These locations allow the operator to save previously logged data and prevent overwrite by another logging activity. Six of the archive areas are designed to hold 15 minutes of 224Kb data, and the seventh area is dedicated to protecting five minutes of the 1.544 Mb/second recorded data. An area has been allocated to provide directory information to the operating programs. The hardware allocation is designed to reduce time-consuming disk head movement and to reduce file segmentation.

To begin a logging operation, the user selects either a Nascom or serial port. For a Nascom port, the software identifies delimiters that may be set by the operator or left blank. This allows the system to identify selected blocks on a port that may be receiving multiple data streams from multiple sources. Start and stop times are selectable, otherwise, all blocks received, which meet the criteria, are logged. If the logging operation is at a T1 rate (1.544 Mbytes), all other logging operations are canceled to dedicate all the system resources to the high rate operation. If a serial port is chosen for the logging operation, delimiters are not available and the system will promptly begin logging all data that arrives at that port. A message in the status area is provided to indicate when the active area is in danger of being overwritten. This allows the operator to write the data to an archive area before it is lost. The status area also briefly indicates which ports are logging and if any CRC errors were received at that input. The main menu provides a short display of all sessions in the high/low speed areas and a quick directory of the archive locations. A selection of one of the sessions or locations will provide a more detailed description including the delimiter parameters and the exact block times within the session.

The playback operation simply places data from either the active or the archive areas directly to the MCP or on the system Ethernet for use at the MOSA workstations. After selecting a session ID or a specific archive area, the operator selects a port and the recorded start/stop times of the data to be transmitted. The playback can be set for transmission at the original rate of the data, the maximum rate for the port, or a user defined rate. For Nascom blocks, the option is available to use an external or an internal clock. If an internal clock is selected, several choices are

provided or the user can specify the rate. Nascom blocks can also be selected by header parameters (or delimiters) during playback. The status window will update to reflect the configuration for the playback with a continuous monitor until the activity is complete.

To delog the data, a separate program is initiated. This program is again written in "C" language. The archive or active area is selected and the recorded start/stop times identified. For Nascom blocks, delimiters are available. The operator has the choice of reviewing the data on the screen, sending it to the printer, or both. The printing and display format is selectable between hexadecimal, decimal, or octal. A delog to the screen will display the configuration at the top of the screen and the blocked data below. The blocks can be stepped through manually or allowed to update at the system rate. Printouts are designed to ensure that a full block is displayed on one page for operator ease of analysis.

MOSA FILE SERVER

The MOSA File Servers are Macintosh Quadra 950 computers running the 4D Client application. The server is accessible remotely from any of the workstations. Each server is independent, but data files can be copied by the operator from one server to the other for backup purposes. The server stores databases and program lookup tables. The operator has the ability to review the databases and in specific instances update the information. To access these databases a server is selected, and the server will prompt the operator for a limited access password. The access level is determined by the system administrator. Once logged to the server, the port status window indicates the activity of the server serial port that is receiving the acquisition data or

teletype. The server status window annotates the number of messages received according to message class. These two windows provide evidence of the server's current standings. Under the database menu are the lookup tables and databases. Each of the database menu items allows the user to create, edit, review, and delete by record (according to access privileges).

The database concepts are particularly important for the tracking program. The tracking acquisition data is automatically stored into a database by way of the serial port. The outdated messages are deleted according to the same criteria utilized by the NASA Ground Network tracking systems documented in the STDN 724. This database allows the user to update acquisition messages according to the needs of the operational environment. As a result, the system allows a retransmission of an acquisition message for purposes of exercising the station tracking and acquisition procedures and configurations, along with generating and transmitting test acquisition data.

The lookup tables maintain information for the MOSA workstation software. The site geodetics, spacecraft identifier/vehicle identifier, station mnemonics, teletype routers, and telemetry configuration parameters are all set values that are provided in lookup tables instead of requiring the operator to enter this data each time it is needed.

The server also stores incoming teletype administrative messages. This database is utilized by the operators to retrieve Briefing Messages, Documentation Change Notices (DCN), Software Support Instructions (SSI), Interim Support Instructions (ISI), and many other operational messages. The

teletype is distributed into relevant categories for easy access by any operator.

Operational documentation is also stored on the server and updated with a word processor as changes are received. Therefore, the latest information is always available by accessing the systems server.

In closing, the MOSA has been designed to enhance NASA's required support to the Ground Network system. The design will easily transition into normal system sustaining. In most cases the actual operators will be able to add functionality for new standardized missions. Though non-standard formats will require software and hardware modifications depending upon specific mission requirements.

REFERENCES

Mission Operations Support Area Detailed Requirements Document, latest revision.

STDN MOSA Data System Design Document 532-SDD-MOSA Nov, 1992.

Mini-NOCC Interim System User's Guide, August 1992.

Mission Operations Support Area User's Guide, August 1993.

Mission Operations Support Area (MOSA) Data System Release 1 Design Review, November 1992.

**SPACE NETWORK SCHEDULING BENCHMARK:
A PROOF-OF-CONCEPT PROCESS FOR TECHNOLOGY TRANSFER**

Karen Moe
NASA/Goddard Space Flt. Center
Code 522
Greenbelt, MD 20771
(301)286-5998, FAX (301) 286-1768

Nadine Happell
Stanford Telecommunications
1761 Business Center Dr.
Reston, VA 22090
(703)438-8028, FAX (703) 438-8112

BJ Hayden
NASA/Goddard Space Flt. Center
Code 534
Greenbelt, MD 20771
(301)286-7307, FAX (301) 286-7911

Cathy Barclay
AlliedSignal Tech. Service Corp.
7515 Mission Drive
Lanham, MD 20706
(301) 805-3221, FAX (301) 805-3228

Summary

This paper describes a detailed proof-of-concept activity to evaluate flexible scheduling technology as implemented in the Request Oriented Scheduling Engine (ROSE) and applied to Space Network (SN) scheduling. The criteria developed for an operational evaluation of a reusable scheduling system is addressed, including a methodology to prove that the proposed system performs at least as well as the current system in function and performance. The improvement of the new technology must be demonstrated and evaluated against the cost of making changes. Finally, there is a need to show significant improvement in SN operational procedures. Successful completion of a proof-of-concept would eventually lead to an operational concept and implementation transition plan, which is outside the scope of this paper. However, a high-fidelity benchmark using actual SN scheduling requests has been designed to test the ROSE scheduling tool. The benchmark evaluation methodology, scheduling data, and preliminary results are described.

Background

The concept of flexible scheduling has been proposed to help meet the Space Network's (SN) anticipated increase in mission support in the late 1990's. The goal of flexible scheduling, which is described in the next section, is increased resources utilization with less manual effort. If SN utilization could be increased 10%, about an additional three service hours per day would be available on each TDRS single access (SA) antenna. This increase provides a total of 24 extra service hours per day, given four operational TDRSs. Scheduling studies have indicated that the flexible scheduling approach will result in increased utilization even if only some customers specify flexibility. ¹ Furthermore, designers of many upcoming missions have indicated a desire to utilize flexible scheduling concepts with the SN. ² Another benefit of flexible scheduling is the reduction in effort required for both customer and Network Control Center (NCC) scheduling operator. ³ For these reasons, the Networks and Data Systems Technology Divisions undertook a detailed proof-of-concept activity to evaluate the flexible scheduling technology as implemented in the Request Oriented Scheduling Engine (ROSE). ⁴

A high-fidelity benchmark has been designed to test the ROSE scheduling tool. This benchmark uses real SN scheduling requests and then modifies them into flexible requests for those customers

who can take advantage of flexibility. The benchmark evaluation methodology and scheduling data are described, as well as how the ROSE tool was tailored and used in the proof-of-concept required by NCC operations. ROSE is a generic scheduling engine and was augmented by the development of two algorithms designed for NCC-type operations, including the lookahead algorithm currently in use in the NCC. Also, a usability test has been defined to specifically test the scheduling system user interface for supporting NCC operational scenarios. Preliminary results of the benchmark functional performance testing are presented.

Flexible Scheduling Request (FSR) Concept

The FSR concept is a candidate for the future Mission Operations Center (MOC) interface to the SN. The concept has evolved over the years based on experience in mission operations in scheduling both spacecraft activities and shared space network services.^{5,6,7,8} The flexible request approach represents a major change in operations concept. Today each customer submits (and resubmits) requests for specific TDRS resources and receives yes/no responses. A large percentage of the rejected requests in the current system are resolved by exercising the users' flexibility through manual coordination.⁹ Alternatively, requirements for space network service requests can be specified in an FSR featuring:

- Flexibility - variable start times, duration, or optional resources
- Repeatability - number of service repetitions and their periods
- Alternatives - primary and backup services
- Constraints - orbital events such as orbits, TDRS antenna view periods, spacecraft day, equator crossings, etc., relationships with other services or requests, or calendar events.

In flexible request scheduling, the user considers all service options and codifies flexible service windows in the request. The space network scheduling system then has more information upon which to base scheduling decisions, increasing the likelihood of successfully satisfying the request. The format for this new scheduling information may be an extension of the Schedule Add Request (SAR) or a new language-based interface.

A key benefit of the FSR concept is the shift of a significant conflict resolution effort from humans to computers. The FSR operations concept minimizes request-response iterations between the network scheduling system and the customer since multiple events can be scheduled from a single request (using repeatability specifications). Also, backup events can be identified and substituted in cases where the primary service is unavailable. The FSR concept supports automated conflict resolution strategies, since tolerances in start times and duration are provided. More events are scheduled, supporting more effective resource utilization. The time to generate a week's worth of schedules can be reduced to hours instead of days.

Proof-of-Concept Evaluation Criteria

For NCC Operations, the goal is for a scheduling tool with built-in flexibility to support an evolving and diverse mission support load without increasing operational complexity and cost. The ROSE scheduling tool is proposed as such a tool. NCC Operations developed a high-fidelity benchmark as a criteria against which to evaluate the scheduling tool. This benchmark must prove that the ROSE scheduling tool performs at least as well as the current scheduling system while providing a significant improvement in SN operational procedures. The SN schedule produced by ROSE must be at least as good in terms of fulfilling customer requests as that produced by the current system. In addition, the process by which the final schedule is produced must show a measurable improvement over the current process, including processing time. The human intervention required by the current process is predicted to be the true bottleneck in scheduling customers in the late 1990's time frame. Therefore, the most important evaluation criteria for a new scheduling tool is the improvement that it can provide to the overall scheduling process in reducing the time consuming human interchanges.

Since NCC Operations will emphasize the scheduling process improvement in evaluating any new scheduling tool, the criteria against which ROSE will be measured consists of computer processing time and manual intervention involved in the total scheduling process. The analysis must go beyond a comparison of computer processing time for a single schedule period based on priority processing. It must be inclusive of the human and computer interfaces between the scheduling personnel at the NCC and those at each customer scheduling facility. The benchmark effort has been, and will continue to be, an effort to determine these measurements.

ROSE Benchmark Proof-of-Concept

The approach for the proof-of-concept involves two phases. In the first phase, the goals are to:

- Perform a high level assessment of ROSE forecast scheduling ability compared to the Network Control Center Data System (NCCDS)
- Verify that ROSE allocates resource appropriately, and
- Compare the computer run times required to generate a forecast schedule prior to manual conflict resolution.

Although the NCC scheduling functions involve additional capabilities, (e.g. request validation, schedule dissemination) resource allocation is the primary function and clearly the most complex. Therefore, our efforts focus on that capability. The resource allocation verification in conjunction with the ROSE usability testing will provide the basis for evaluation of ROSE as a scheduling tool

from an NCC perspective. This phase will be referred to as "Phase I: Schedule Comparison and Resource Allocation Verification."

In the second phase, the emphasis is on demonstrating improvements in operational procedures and evaluating ROSE from a SN customer perspective. ROSE has potential for significantly reducing the time required to perform the current NCC forecast schedule generation process because of its support of flexibility. This benefits both NCC and customer operations. In this phase we will quantify the reduction. Also, since ROSE capabilities involve much greater degrees of flexibility than the current NCC, it is important that the customer understand how they can effectively use these new flexibility options. The customers' evaluation of the resulting schedule is key to the overall assessment of ROSE. This phase will be referred to as "Phase II: Procedure Improvement and Flexibility Analysis."

For Phase I we describe the process, the environment in which the process was performed, the schedule and related data utilized, and finally summarize preliminary results. Space doesn't allow a similar description of Phase II, however we will highlight the new features and indicate status.

PHASE I - Schedule Comparison and Resource Allocation Verification

The Process. The key drivers for devising the resource allocation verification process are time and realism. Our goal is to perform a high level comparison of schedules and computer run times, as well as to check that ROSE schedules SN resources without conflict. An in-depth detailed verification would take on the order of several months and require skilled test personnel. This type of testing will be performed in more formal testing phases if ROSE is selected as an NCC scheduling system. Realism is key because we want to focus the evaluation on the most common types of resource conflicts encountered, while still ensuring all resources are allocated properly. The data flow for the Phase I process is illustrated in Figure 1. Shaded boxes indicate completed activities at the time of publication.

Both drivers can be addressed by using operational SARs and related data for an NCC forecast week and submitting them to both the NCCDS and to ROSE. The resulting schedules will be compared in terms of number of events scheduled and minutes of support scheduled. Currently, the operational SARs express flexibility in event start time and SA antenna, therefore, it is possible to generate different conflict-free schedules. However, one schedule may better satisfy the SN customers.

While this comparison provides a foundation for evaluating schedules and run times, an additional step is required to verify ROSE resource allocation. The ROSE scheduled events will be formatted

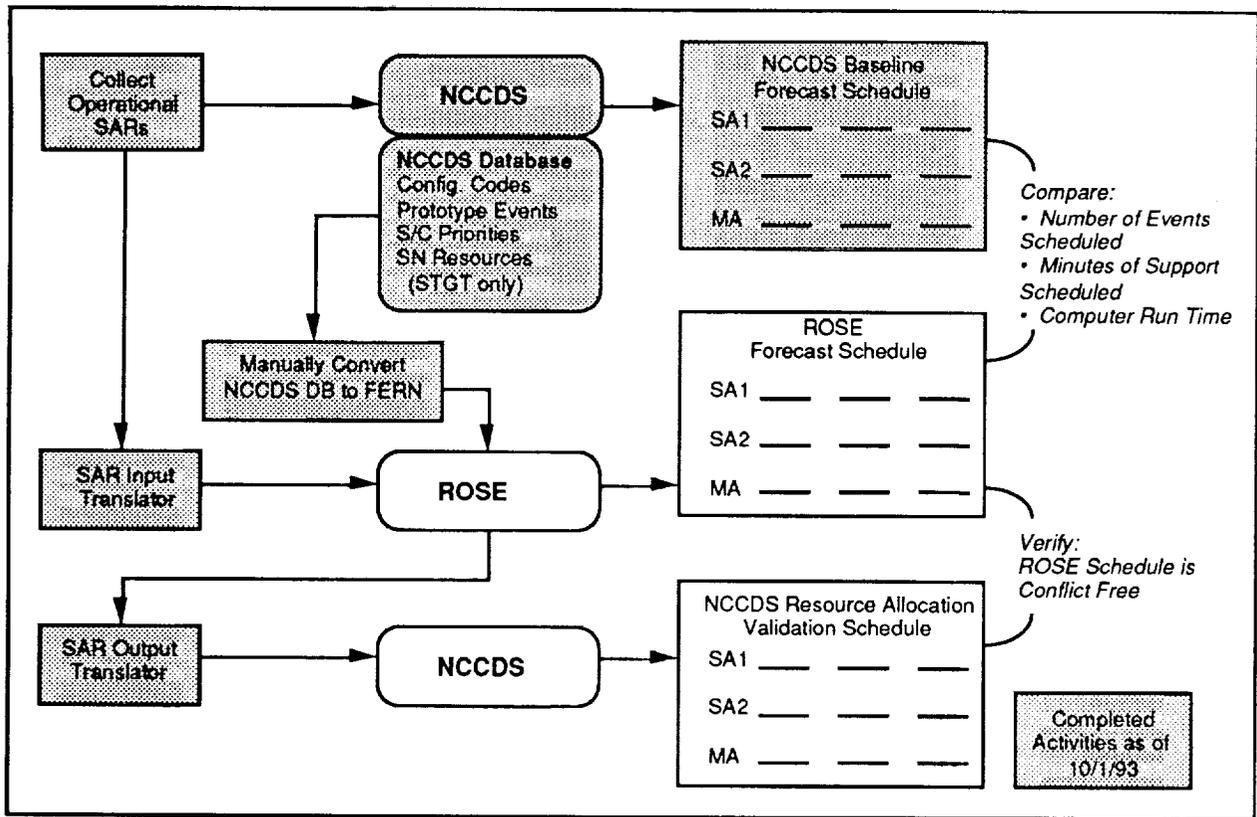


Figure 1. Phase I Process Overview

into specific SARs, without start time flexibility, and submitted back to the NCCDS. Another NCCDS forecast schedule generation run will be performed to determine if ROSE scheduled any requests in conflict based on NCCDS resource allocation rules.

The Environment. Utilizing actual SARs requires that the entire process be performed in a classified environment. The NCCDS baseline schedule and the resource allocation verification schedule run will be performed within the NCC. A SUN Sparcstation will be installed in the NCC to support the ROSE benchmark evaluation effort. The results of the evaluation will be presented in an unclassified manner.

To provide an accurate comparison of the NCCDS baseline schedule and the ROSE generated schedule, the comparison is made for running all SARs together in one schedule generation run. The NCCDS baseline schedule prior to any manual conflict resolution is the result of that run.

In the NCCDS, the resources utilized in the schedule run are dependent on the ground terminal that supports the available TDRSs. Since the time frame in which a new NCC scheduling system would be implemented is likely to be after the Second TDRS Ground Terminal (STGT) is operational, all TDRSs will be assigned to STGT for all of the Phase I schedule runs.

Schedule and Related Data Collection

It was necessary to carefully choose and coordinate the SARs, configuration codes, prototype events, and spacecraft priority list for all of the tests. To obtain the operational SARs, a forecast week was selected that included a shuttle mission to address a significant scheduling workload. Coordination with the SN customers was necessary in gathering mission scheduling data. The SARs were logged by the NCCDS Test System (NTS) for the test. Compatibility tests were performed between the NTS and ROSE to ensure that they could reliably exchange the SAR data. The NTS will be used to extract the SARs and to submit the ROSE generated SARs to the NCCDS for the resource allocation verification step.

Configuration codes and prototype events are specified in the SARs. Copies of these were provided to ROSE to ensure that the same resources are requested in all schedule runs. The same spacecraft priority list will also be used in all schedule runs in accordance with operational procedures. The same NCCDS database which contains the configuration codes, prototype events, and spacecraft priority list will be used for the NCCDS schedule run to verify ROSE resource allocation. The NCCDS baseline schedule was generated using this database, after all validated SARs were received for the selected forecast week.

The boundary between the active and forecast period was also addressed. Some of the active period events start late in the day on the last day of the active period and overlap into the forecast period. These events were included in the forecast period schedule data collected for the tests.

Data Preparation for Input into ROSE

The first challenge from the ROSE perspective was deciding how to represent NCC information in ROSE's Flexible Envelope Request Notation (FERN).¹⁰ This information falls into the following general categories: resources for allocation, scheduling ground rules, and requests. Scheduling ground rules include specifications for setup buffers on resources (i.e., the time required between uses of a resource), duty factors on the Multiplexer/Ddemultiplexer (MDM) and Statistical Multiplexer (Stat Mux), and restrictions on the choice of TDRS antenna within an event.

Scheduling Resources. In order to schedule the SN, both communication services and TDRS antennas must be allocated, although there is a very strong tie between them. For Multiple Access (MA) services, one service equates to one antenna. However, multiple SA services may be scheduled on a single physical antenna, provided all services are for the same customer (since the antenna can point to only one spacecraft at a time). Customers request services. However, once one SA service is assigned to a customer, the remaining SA services on that antenna can not be assigned to any other customer. Therefore in requesting an SA service, the customer is in effect requesting an SA antenna. However, due to equipment failures, not all SA antennas support all

services. The NCC maintains equipment status, and therefore must insure that a customer is assigned to an SA antenna capable of supporting the services requested.

The definition of the resources within ROSE accounts for this close coupling of two different resource types (antennas and services). To do so, the physical antennas were defined as the most primitive resources. Each service was then defined as a pool of physical antennas capable of supporting that service. In order to insure that all SA services for an event were assigned to the same SA antenna, combinations of services were defined as resources that list all of the physical antennas capable of supporting all services in that combination. The customer then specifies a service combination as a requested resource. ROSE assigns the event to a physical antenna listed in the service combination definition. This physical antenna is no longer available to other requests for the time frame requested. However, this strategy does allow multiple services from the same request to be assigned to the same physical antenna.

Other resources, such as interface channels and duty factors, are more independent and made a fairly simple transition into FERN and ROSE. Requests for certain customer interface channels do imply a need to request certain duty factors, however this relationship impacts the original request generation and not the scheduling process. Interface channels are simple capacity resources; customers request a single unit of each interface channel and they are either available or not. Duty factors are consumable/renewable resources, where customers request multiple units corresponding to their required maximum bandwidth. The combined bandwidth of all services from all customers cannot exceed a certain threshold.

Scheduling Ground Rules. The majority of the scheduling ground rules were worked into the resource definitions. The duty factor constraints were implemented as consumable/renewable resources as mentioned above. Resource setup buffers specify the amount of time required between each use of a resource for reconfiguration for the next support. The SN uses two setup buffers for many resources, one where the next use of the resource is in another event (external buffer), and one within the same event (internal buffer). FERN has an option within the resource definition for specifying a minimum gap between each use of the resource, but is incapable of expressing internal event buffers. Since all external buffers are equal to or slightly larger than setup buffers internal to an event, only the external buffers were implemented. Not using internal buffers was insignificant to the benchmark, since the likelihood of an event containing back-to-back use of a resource within the internal buffer is improbable, given the experience of current SN customers.

There are additional ground rules that specify that all services within an event must be on the same TDRS, and that if a service stops and restarts within an event, the same antenna shall be used for each instance of the service. These restrictions were incorporated into the scheduling algorithm.

The NCC Lookahead Algorithm. The NCC lookahead algorithm uses a conflict avoidance strategy. The basic principle is to examine the placement options of an event, and schedule it in the spot that is least likely to create a conflict with a lower priority pending request. Figure 2 illustrates the logic flow as implemented in ROSE.

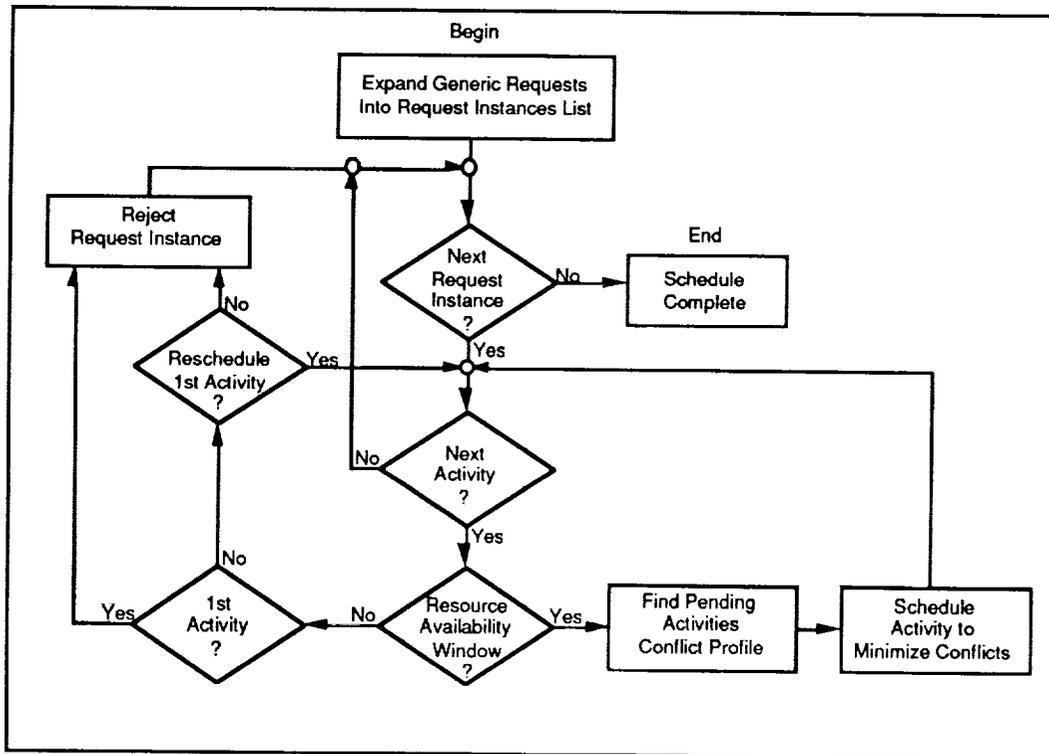


Figure 2. Lookahead Algorithm Logic as Implemented in ROSE

The lookahead algorithm implemented in ROSE, has some subtle differences with the NCCDS implementation. The NCCDS version looks for potential conflicts across services; the ROSE version looks for them across physical antennas. It was determined that this difference would have no impact on schedule outcome. The NCCDS version selects the best location for an event based on the combined potential for conflict across all services in the event. The ROSE version selects the best location for each activity (or service) individually. In flexible requests, it is possible to have more than one activity per event, therefore the best location for the event is not guaranteed. However, for Phase I, where the requests are relatively inflexible, there is a one-to-one correspondence between activities and events and no impact should be seen.

Other differences occur in the minute details of the implementation. These include differences in step size when sliding the start time around within an open window, and differences in the size of the weighting factors in computing a conflict sum for each start time and resource option. In both implementations, potential conflicts with the next highest priority request are weighted more heavily than potential conflicts with the very lowest priority request. Also, potential conflicts on resources that are in higher demand are weighted more heavily than potential conflicts on

infrequently used resources. The ROSE implementation took advantage of a ROSE feature that calculates current resource utilization, so that the resource weights can be adjusted dynamically.

Requests. The SN services are grouped as an event in a SAR. However, FERN structures its requests hierarchically, as shown in Figure 3. The generic structure specifies repetition instructions, the activity specifies a sequence of steps and the duration of each step, and the step specifies the resources that are required for that period.

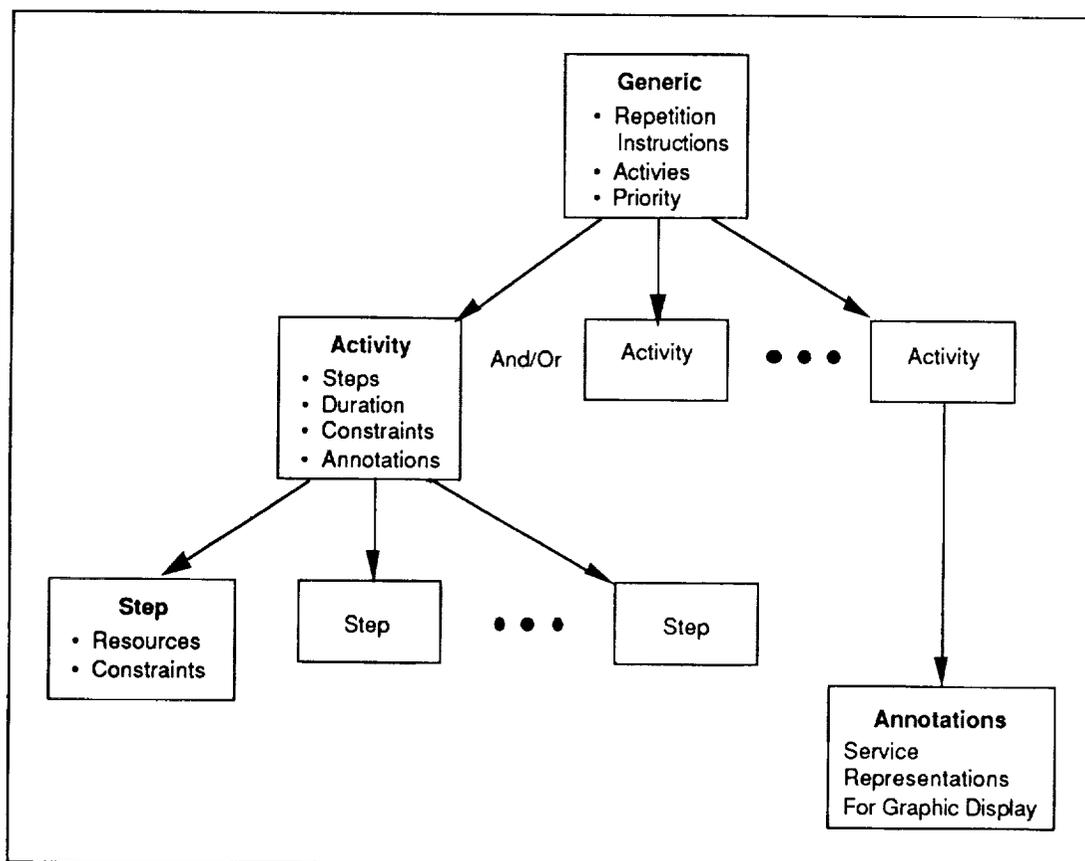


Figure 3. FERN Structure

Steps within an activity are strictly sequential, whereas SN services specified in a SAR typically overlap. Current NCC SARs require the start time of all services to be fixed with respect to one another. This restriction allows the time slicing of the event when converting to FERN. Whenever a service either starts or stops, a new step is defined. Steps then list all resources required for all of the services that are ongoing at that time. For example, an event composed of SSAF, SSAR, and Tracking services, would be represented by an activity with four steps as shown in Figure 4. SA services are time sliced within the activity, however, they always follow the same order (i.e., forward, return, tracking). Thus the combination of FERN activities and steps represent current SN events with services being time sliced among the steps.

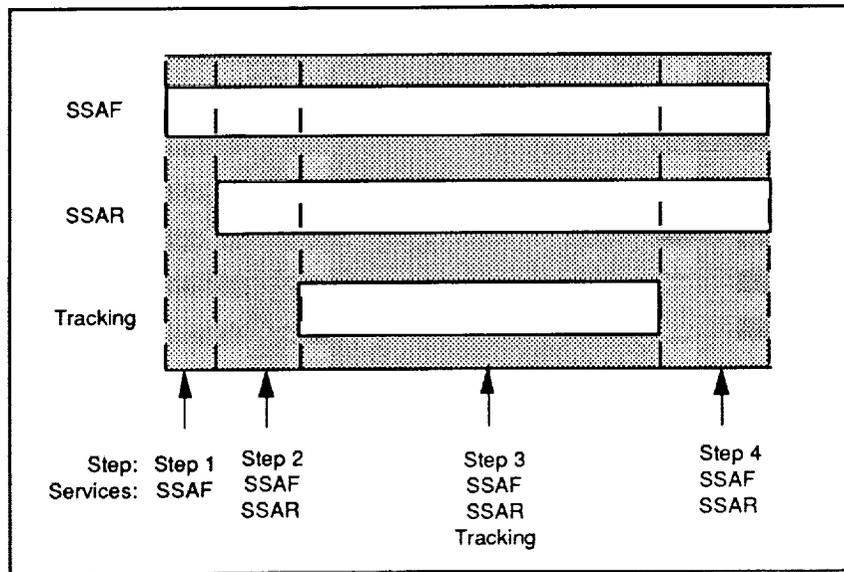


Figure 4. Time Sliced Event Represented in FERN

SAR Input Translator. A SAR translator, was written in the Unix interpreted language AWK. This translator reads in the SAR and reformats it into FERN using the time slicing methodology. Since current SARs are non-repeating, the FERN requests will also be unique, with one generic composed of one activity for each SAR. Each generic and activity is labeled by customer and SAR message id. Each step name lists the customer, message id, and all configuration codes supported during that step. Another FERN structure called an annotation, describes each configuration code. The information in the annotation is not used by the scheduling logic, but only by the display system. Annotations permit services to be displayed as a whole, and not time sliced into multiple steps.

The configuration codes specified in the SAR contain important information concerning requested resources. Since configuration code definitions are in the NCC database, a configuration code database was built for the SAR translator. The SAR translator database is significantly smaller than the NCC configuration code database since it includes only that information required to support resource allocation and only contains those configuration codes actually used during the test week. The SAR translator also references the list of the mission priorities. These are the default priorities inserted into the FERN requests. However, some users had several critical requests that were given a higher priority. These request priorities will be manually modified in the FERN requests. After all SARs are run through the SAR input translator, separate FERN request files will be created for each customer. The only other file needed for the Phase I test is the file describing the SN resources to be allocated. A schedule can then be run using the lookahead algorithm.

SAR Output Translator. The ROSE output schedule is to be submitted back to the NCCDS to verify that ROSE does not inappropriately schedule any requests. The ROSE output schedule must be translated back into the SAR format. Another translator has been developed for this purpose.

The resulting SARs should express no flexibility at all, since they represent scheduled events assigned to specific resources. However, the SAR format does not directly express all required resources; some resources are specified in the configuration codes. The SAR output translator determines the configuration codes used by each event and references them in the output SAR. Unfortunately, most configuration code definitions express flexibility in the antenna choices. If the NCCDS were to choose a different antenna than what ROSE actually chose, the change in one event could cause a ripple effect and produce a different schedule. However, it appears that the NCCDS allocates antennas in numerical order. The resources in FERN can be listed in a similar manner, so that the search pattern in both systems should be the same, and result in the same assignments. If this strategy fails, however, new configuration codes must be defined with very specific resources, and the SAR output translator can be set to reference these new codes.

Phase I Preliminary Results and Status

As of October 1, 1993, we have completed the NCCDS baseline schedule and the description of the result follows. The ROSE schedule is expected to be run later in October after the host workstation is completely installed and procedures are completed for handling classified data.

The forecast week selected was September 13-19, 1993 (256/00:00:00 - 262/23:59:59 Z). The seven day operational forecast process for this week took place beginning on August 30, 1993. We extracted the SARs on August 31 and performed the NCCDS baseline schedule run on September 1st. The NCCDS baseline schedule run included 1028 unclassified SARs and took just over 45 minutes to complete. We measured the primary and secondary resource scheduling separately. Primary resources are the SA and MA Forward (MAF), and start time tolerances are used to schedule these resources. The STGT era secondary resources used in this schedule run were the MA Return (MAR), customer interface channel, MDM bandwidth, and Stat Mux bandwidth.

Table 1 summarizes the initial result of the NCCDS schedule run for the unclassified customers on STGT by order of spacecraft priority before any conflict resolution was applied. The declined SARs were due to conflicts on the following resources:

- 90% - SA or MAF Conflict

- 10% - MAR Limitation

- <1% - User Interface Channel Conflict (one HST request declined)

The set of declined SARs attributable to the MAR limitation are due to the difference between WSGT and STGT resources. TDRS spare was assigned to the third equipment set at STGT which does not support MA, hence, SARs for TDRS Spare MAR were declined. Therefore, in actuality nearly all the SARs were declined due to SA or MAF conflict. The results indicate that

lower priority customers who specify tolerances in their requests, like COBE, do increase the likelihood of getting their requests scheduled.

SN Customer	% SARs w/Tolerance	SA Flexibility	% SARs Scheduled	% SARs Declined
GRO Critical	0	Yes	100	0
UARS Critical	0	Yes	100	0
STS - 51	0	No	99	1
HST	< 1	Yes	64	36
GRO	0	Yes	74	26
TOPEX	92	Yes	67	33
EUVE	99	Yes	66	34
UARS	95	Yes	55	45
COBE	70	Yes	89	11
ERBS	99	Yes	54	46
Total			79	21

Table 1. NCCDS Forecast Baseline Schedule Statistics Prior to Conflict Resolution

The NCCDS baseline schedule also had the following computer run time statistics (mm:ss):

Primary SN resources: 03:06
 Secondary SN resources: 42:11
 Total: 45:17

Similar statistics will be generated for ROSE under Phase I testing in October. Upon completion, we will compare the NCCDS and ROSE schedules and computer run times, and verify that ROSE did not schedule any conflicts. Finally, we will analyze the remaining SARs from NCCDS which were not resolved during the manual conflict resolution process, and any SARs from ROSE which do not get scheduled.

PHASE II - Procedure Improvement and Flexibility Analysis

The Process. As in Phase I, key drivers for devising the process for procedure improvement involve time and realism, so again we will compare NCCDS and ROSE schedule runs using operational SARs and compare the number of events/minutes scheduled and the computer run times. The configuration codes, prototype events and the spacecraft priority list used in Phase II will be the same as that used in Phase I. However, in this phase we will also compare the times to

perform the entire procedure to create a forecast schedule including conflict resolution. Figure 5 illustrates the Phase II process.

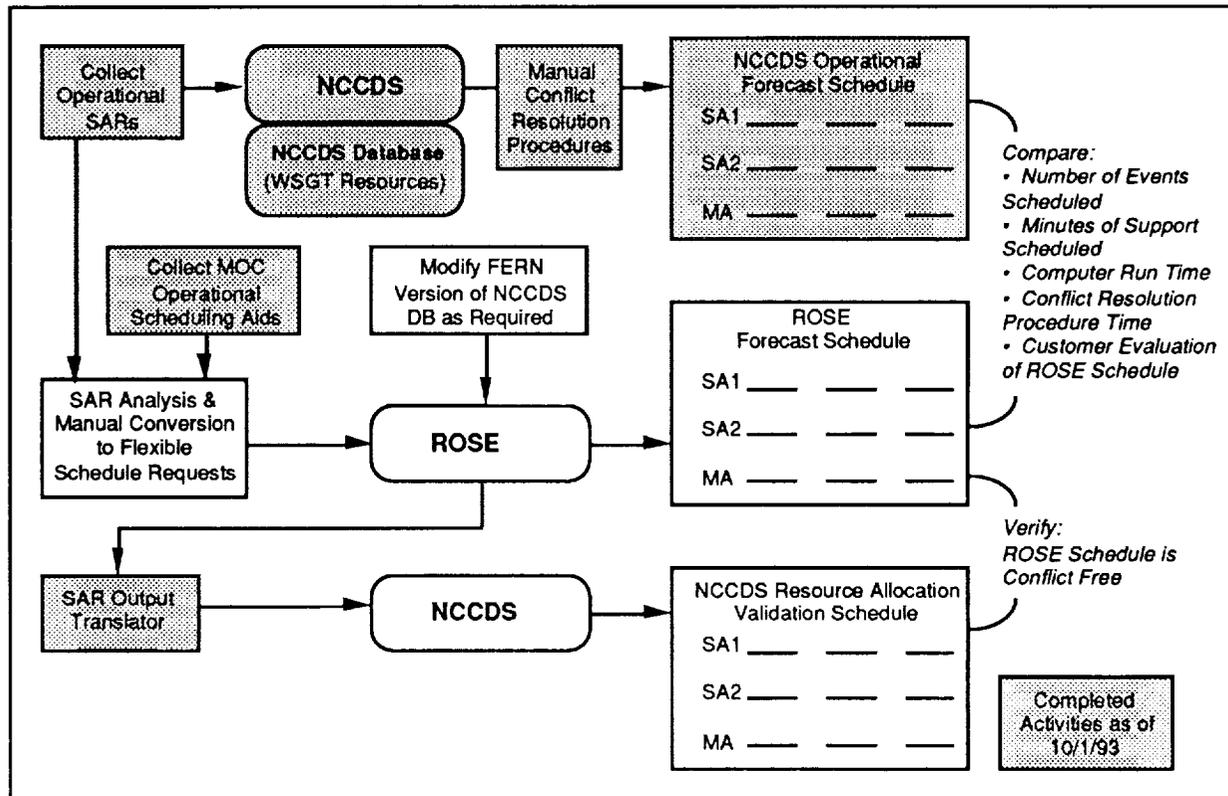


Figure 5. Phase II Process Overview

The NCCDS baseline schedule run for Phase II was performed by the operations personnel. The week selected is the same week as that used in Phase I. When coordinating with each customer for Phase I, we also requested that each customer save the scheduling data used to generate their SARs for support of Phase II. In addition, operational schedule run data, observation notes, and conflict resolution notes were saved. This data will be used to analyze the additional types of flexibility available to the customer.

The additional types of flexibility will be specified as Flexible Scheduling Requests (FSRs) in the FERN language. ROSE will utilize the FSRs to generate a forecast schedule that will be compared to the NCCDS operational schedule run on WSGT resources. At first, it may seem inappropriate to compare an NCCDS schedule run on WSGT resources to a ROSE schedule run on STGT resources. However, it turns out the differences between the resources have minimal effect on the computer and procedure run times. As for the customer perspective, we expect the differences in WSGT and STGT resources to have minimal effect on the use of flexibility. Of course, the difference in resources will be considered when comparing NCCDS and ROSE schedules in terms of number of events and minutes of support scheduled. Finally, ROSE will again generate specific SARs that

correspond to the events scheduled using the FSRs. The specific SARs will be scheduled using the NCCDS to provide additional resource allocation verification as with Phase I. The physical environment is the same as Phase I.

User Scheduling Data Collection

In order to translate the original SARs into flexible requests, we need to understand the customers' true flexibility options. We need to understand how they chose where to schedule the requests initially, and how they chose what conflict resolution options were acceptable to them, and which options were preferred over others. We also need to collect any data, such as user antenna views (UAVs), that they may have used in making those decisions. We felt that this data should be collected as soon after the test week as possible, so that the activities were still fresh in the customers' minds, and that files and tapes were not overwritten or deleted.

For this data collection process, we visited all of the GSFC Mission Operations Center (MOCs) and interviewed their scheduling personnel. We also requested that they complete a short questionnaire concerning the conflicts they encountered during the test week, and how they were resolved. By personally visiting each customer, we were able to gain a very clear and detailed understanding of the customer's side of the process for the test week, as well as collect the scheduling aids (e.g., view period data).

The types of flexibility that different customers may have that cannot be expressed in the current SAR are the ability to accept:

- Shorter service duration
- Any TDRS as long as it is in view
- Service start time tolerance
- Wider event start time tolerance
- Different type of service (MA vs SA)
- Moving the contact to another orbit
- Periodically repeating an event

One of the key flexibility concepts is SA service start time tolerance with respect to MA services within same the event. A previous study showed that 71% to 79% of Hubble Space Telescope's conflicts could be resolved using this type of flexibility.¹¹ The time sliced event representation strategy discussed under Phase I allows flexibility in the relative start times of the different services.

Preparing the FSRs. For Phase II, the request representation we settled on was to have one activity for each physical TDRS antenna requested (versus one activity for each event in Phase I). The generic data structure allows conjunctions of activities, therefore one generic would specify that the

activities in an event must all be scheduled as a unit on the same TDRS. This representation strategy allows flexibility in the relative start times of the different activities.

Most FSRs will be generated manually based on an interpretation of the data collected from the MOCs. For most customers this effort is not expected to be excessive, since they can use the repeatability factor of flexible scheduling, and one FSR can replace many SARs. For those customers who would not use flexible scheduling, the SAR translator can regenerate their specific requests. The SAR translator will likely be modified to also support flexible non-recurring requests. After the FSRs have been created, the customers will verify that they represent acceptable conflict resolution options in the correct order of preference.

Phase II Preliminary Results and Status

The NCCDS operational schedule with WSGT resources and including manual conflict resolution was completed during the forecast week of August 30, 1993, and the results were collected. The number of events at the end of the operational NCCDS schedule run was higher than at the beginning of the week for the initial forecast schedule. This discrepancy is due to additional (late) SAR submissions during the forecast week, and conflict resolutions that include splitting one event into two.

Based on the customer interviews, there were no resolution options exercised that week that could not be expressed in an FSR. Some requests simply could not be satisfied as there were no acceptable resolution options. We are optimistic that ROSE will be able to find conflict resolution options for all requests that were eventually scheduled. The MOCs will also be asked to judge if the options that ROSE found were as good (or maybe even better) than those found manually. We hope to be able to estimate the number of staff hours saved by using flexible scheduling over manual conflict resolution. Table 2 shows the results for the operational schedule after conflict resolution. The column entitled "% Increase over Baseline" shows the additional percentages of requests scheduled over the first run of the forecast schedule before any manual conflict resolution was completed.

Generation of the NCCDS operational schedule had the following computer run times (mm:ss):

Primary SN resources:	06:30
Secondary SN resources:	<u>118:30</u>
Total:	125:00

In addition, 60 hours of forecast schedule operator support time (including wait time for MOC responses to recommended resolutions) were allocated to the manual conflict resolution procedures for the week. The NCCDS preliminary results from Phase II, as compared to the baseline schedule, illustrate the significant increases in scheduled support due to the manual efforts of the forecast

operators and customer scheduling personnel. The goal of the flexible scheduling concept is to automate conflict resolution in the schedule run. Similar data will be collected for ROSE during Phase II testing.

SN Customer	% SARs w/Tolerance	SA Flexibility	% SARs Scheduled	% Increase over Baseline
GRO Critical	0	Yes	100	0
UARS Critical	0	Yes	100	0
STS - 51	0	No	100	1
HST	< 1	Yes	99	35
GRO	0	Yes	94	20
TOPEX	92	Yes	100	33
EUVE	99	Yes	98	32
UARS	95	Yes	91	36
COBE	70	Yes	99	10
ERBS	99	Yes	86	22
Total			97	18

Table 2. NCCDS Operational Forecast Schedule Statistics After Conflict Resolution

Lessons Learned to Date

The ROSE evaluation exercise began in May 1993, and in a relatively short time frame we have established a methodology and collected necessary test data for both phases of testing. This same methodology could be adapted to similar technology evaluations by other operational systems or by the NCC for other proposed enhancements.

It was necessary to coordinate the collection of data both in the NCC and the MOCs for the selected forecast scheduling week. The evaluation team held regular status meetings and established close contacts with scheduling and database operations personnel which had several benefits. Many detailed but critical points were uncovered and resolved during the process. Everyone maintained an eye on the evaluation goals and stayed well informed. Close cooperation was needed between government and contractor personnel, as well as between organizational elements. However an end result was the high level of interest in the process and results.

In the process of understanding the scope of scheduling in the NCC, we learned about subtle details of the database with its embedded scheduling ground rules, and the importance of finding an appropriate representation for SARs. The initial conversion of the NCCDS database into FERN for

ROSE input took about a staff month of effort for both analysis and implementation. Representing events as overlapping services had to be carefully addressed in ROSE as there were alternative implementation approaches. Once determined, the implementation of the SAR translator was fairly straight forward, taking only a couple staff weeks.

The lookahead algorithm developed for ROSE was based on the NCC lookahead algorithm. However, the ROSE algorithm takes into consideration the flexibility options which are not part of the NCC algorithm. Further testing and some modifications will be required to fully validate the ROSE algorithm, however the tolerances and open selection capabilities in the current SARs have been successfully tested. Approximately 9 staff months of effort went into the redesign, Ada implementation, and testing of the ROSE algorithm to date.

As we prepare to actually run the validation benchmark test on ROSE, we anticipate that additional testing may be required, in spite of successfully testing individual components of the benchmark. However, we feel we have developed a robust methodology that will be able to adapt to the new lessons we will learn.

Conclusions and Future Work

Technology transfer to operational elements involves the necessity of having to maintain support while upgrading to a new way of doing business. The NCC requires a proof-of-concept benchmark, such as the one we have described, in order to verify the value of proceeding with the time consuming task of transitioning into operations. In addition to the validation of ROSE for producing SN schedules, a usability test of the ROSE user interface is also in process. The ROSE usability test is designed to verify that the interaction techniques provided by ROSE fully support the scheduling tasks performed by scheduling operators. Thus direct evaluation by NCC Operations personnel will assess ROSE utility and drive the need for modifications.

If a new technology proof-of-concept such as ROSE is successfully demonstrated in the testing process the next step is to develop a methodology for transition into operations. A complete operations concept is required, addressing both the NCC and MOC roles and operational procedures in the Flexible Scheduling approach. Since a major change to the NCC requires changes of the customer community, representatives of the SN and the MOCs have been working on a joint paper, "Space Network Flexible Scheduling Enhancements", to identify desired enhancements which will be validated in Phase II. Finally, detailed plans for integration of the new technology into the NCC, and plans for formal acceptance testing will be required.

Acknowledgments

The authors would like to thank the following people whose support was crucial to accomplishing this effort: Tony Maione, Candace Carlisle and Lynn Myers of the NCC Project Office who endorsed the evaluation concept and supplied the physical environment for the tests; Les Wentz of the Mission Operations Division who supported the coordination efforts with the MOC; Stuart Weinstein of Loral AeroSys, the ROSE developer who solved many ROSE interface questions; Ron Cook and Karen Streu of Stanford Telecom. who developed the NCC lookahead algorithm for ROSE; and from AlliedSignal Tech. Services Corp., John Groom/NCC Database Manager, Steve Edwards, and forecast analysts Fred Daves, Pat Quimby and George Kellenbenz who supported the collection of the test data.

References

1. "SN Mission Drivers for Flexibility" presentation to the 4th SN/STGT Integration Workshop, page L-14, May 20, 1993.
2. Study notes from SN Customer Scheduling meetings, CSC, Feb. 1993.
3. "SN Customer Scheduling Flexibility Analysis" presentation to the 4th SN/STGT Integration Workshop, page L-23, May 20, 1993.
4. "Request-Oriented Scheduling Engine (ROSE) User's Guide", S. Weinstein/Loral, DSTL-93-001, NASA/GSFC Contract NAS5-31478 and NAS5-31500, May 1993.
5. "An Expert System for Scheduling Requests for Communications Links between TDRS and ERBS", D. McLean, R. Littlefield, D. Beyer, Telematics and Informatics, Vol. 4, No. 4, 153-261, 1987.
6. "Lessons Learned Prototyping Flexible Scheduling for the TDRSS Domain", S. Munns, T. Welden/CSC, D. Crehan/Loral AeroSys and N. Goodman/GSFC, DSTL-91-003, NASA/GSFC Contract NAS5-31500, Aug. 1991.
7. "Space Network Control System/Payload Operations Control Center Flexible Scheduling Concepts", Stanford Telecom., Inc. TR-91-108, NASA/GSFC Contract NAS5-31260, Jan. 21, 1992.
8. "Space Network Control Conference on Resource Allocation Concepts and Approaches", K. Moe/GSFC, Ed., NASA CP-3124, Dec. 1990.
9. "Space Network Flexible Scheduling Enhancements", Stanford Telecom. final draft, TR93102, Sep. 13, 1993.
10. "Flexible Envelope Request Notation (FERN) User's Guide", G. Michael Tong/GSFC, DSTL-93-012, May 1993.
11. "Hubble Space Telescope Scheduling Analysis", Stanford Telecom. final report, TR92021, Nov. 13, 1992.

N94-21333

**Using Manufacturing Message Specification
for Monitor and Control at Venus**

W. Randy Heuser, Richard L. Chen and Michael H. Stockett
Monitor & Control Technology Group
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract

The flexibility and robustness of a monitor and control (M&C) system are a direct result of the underlying inter-processor communications architecture. A new architecture for M&C at the Deep Space Communications Complexes (DSCCs) has been developed based on the Manufacturing Message Specification (MMS) process control standard of the Open System Interconnection (OSI) suite of protocols. This architecture has been tested both in a laboratory environment and under operational conditions at the Deep Space Network (DSN) experimental Venus station (DSS-13). The Venus experience in the application of OSI standards to support M&C has been extremely successful. MMS meets the functional needs of the station and provides a level of flexibility and responsiveness previously unknown in that environment. The architecture is robust enough to meet current operational needs and flexible enough to provide a migration path for new subsystems. This paper will describe the architecture of the Venus M&C system, discuss how MMS was used and the requirements this imposed on other parts of the system, and provide results from systems and operational testing at the Venus site.

Introduction

The Deep Space Network (DSN) operates the Venus facility at the Goldstone Deep Space Communications Complex as a research site to conduct experiments in new tracking techniques, communication technologies, and instrumentation. In response to expanding demand on station resources generated by the new 34-meter Beam Waveguide Antenna, a new monitor and control system was developed and installed based on OSI protocols [1]. As an experimental facility, the Venus site is constantly changing. New instrumentation is moved to the station and tested on a weekly basis. The procedures to support observations or spacecraft activities change daily. However, Venus is also used routinely for radio astronomy observations and the High Resolution Microwave Survey (HRMS). The Station Monitor and Control system was designed to support the constantly changing conditions at the station, while providing reliable support for operational experiments.

Requirements

The basic requirements for Station Monitor and Control were derived from the requirements for monitor and control of the DSN operational facilities. The operational facilities are a collection of computer automated antennas and support equipment which are linked together with a Local Area Network (LAN). The station is operated from a set of central control consoles which support the following requirements:

- 1) Allocation of station resources to provide assignment of subsystems in support of station activities.
- 2) Support data files are received at monitor and control and redistributed to subsystems as required.

- 3) Operator directives entered on monitor and control support the configuration and operation of subsystems.
- 4) Subsystem health and performance is displayed on monitor and control.
- 5) Subsystem event or alarm conditions are generated and reported to monitor and control.
- 6) Monitor data is exchanged between monitor and control and the active subsystems based on negotiated interface agreements.

In addition to these basic requirements, the constantly changing environment at Venus created the following requirements.

- 1) The Station Monitor and Control system must support the addition of a new subsystem without modification to the software.
- 2) The Station Monitor and Control system must support modification of displays and construction of new displays without new software deliveries.

The additional requirements reflect concerns for flexibility and low life cycle costs. The design of Station Monitor and Control makes extensive use of commercial software products with the tools that provide the required flexibility.

Station Monitor and Control Architecture

The OSI based monitor and control architecture enables operation of the Venus site through a single operator workstation.

OSI protocols are used to link the operator workstation and the various station subsystems across the LAN. Commercial software packages based on the OSI Manufacturing Message Specification (MMS) protocol were employed to support all interprocessor communications. A commercial Supervisory Control and Data Acquisition (SCADA) software package was incorporated to support most of the operator interface functions through a graphical user interface (GUI). The extensive application of commercial software packages made it possible to build and install the system in one year at substantial cost savings [2].

There are two aspects to the OSI based monitor and control architecture. The first is the communications architecture which employs all seven layers specified in the OSI Basic Reference Model. Support for the communications architecture resides on all of the station computers. The second aspect involves the structure of the program (or programs) that runs on each of the station computers. This structure is called the software architecture and is tightly coupled to the computer hardware, the operating system, and the computer language employed. Since 98% of the software for Station Monitor and Control was purchased, much of the software architecture was set by the vendors. This discussion is therefore limited to those components used to integrate the commercial products into a working system. Both aspects of the architecture are depicted in Figure 1 and the station network configuration is seen in Figure 2.

Station Monitor and Control Implementation

The Station Monitor and Control system was initially implemented on an Intel 80386, 33 MHz computer running the Interactive UNIX operating system. The computer platform was provided to the project and selection of the software components was governed by the platform. The selection of the UNIX operating

system was made to provide a flexible multi-tasking environment. The platform and operating system directed the selection of the specific MMS/OSI software package. In addition, the platform and operating system led to the selection of a Supervisory Control and Data Acquisition (SCADA) software package called Dexterity. The Dexterity product met all of the basic requirements to support the operator interface and provided an Application Program Interface (API) to support software development. The software development effort was limited to providing a link between Dexterity and the MMS network services.

The MMS software package provided a set of libraries and sample code to build MMS service calls. Using this package and the Dexterity API, the following elements were developed to support the MMS/OSI network services:

- o The **MMS-Dexterity Network Server** provides all of the network services for Station Monitor and Control. The network server is a program that runs concurrent to Dexterity under the UNIX operating system and by UNIX convention is called a daemon.
- o The **Dexterity Interface Program** is used to initiate the MMS-Dexterity network server to take action. The program is initiated to run when an operator uses the mouse or keyboard to trigger some network action.
- o The **MMS-Dexterity Timer Program** runs concurrent to the network server and provides a heartbeat to initiate polling of subsystem status data.

The key and most complex element of the software architecture is the MMS-Dexterity Network Server. The network server is started simultaneously with the Dexterity product and is table-

driven, so changes to the network can be accommodated without changing the network software. The network server establishes the environment to communicate across the LAN by reading a table with all of the network address information. Next, the network server reads a set of tables that identify all of the control and performance parameters used to operate the subsystems. Finally, the network server establishes communications with the Dexterity database.

Once initialized, the network server is ready to support all of the network communication services required to operate the complex from the Station Monitor and Control workstation. Station personal operate the complex through a graphical user interface (GUI) managed by the Dexterity software. The Dexterity software supports operator actions through the keyboard or a mouse. The Dexterity database services the user screens. In addition, the Dexterity product provides a set of tools to build and manage the GUI screens. The basic operation of the station is accomplished as follows:

- 1) Station resources are allocated to support station activities by establishing a connection between Station Monitor and Control and the selected subsystems. Resources are released by concluding the connection. The process of establishing and releasing a connection is accomplished using the mouse.

- 2) Subsystem health and performance is read by or reported to Station Monitor and Control using the MMS variable read service or the information report service. Changes in subsystem status and performance are echoed on Station Monitor and Control using the same MMS services.

- 3) Operators control a subsystem by entering configuration parameters through the GUI into the Dexterity database. The operator action triggers the Dexterity Interface Program to execute and pass the selected parameters to the Dexterity Network Server. The network server extracts the selected data from the Dexterity database and writes the data across the network to the subsystem.
- 4) Support data files are received by Station Monitor and Control from JPL across the NASA Science Internet. Operators distribute support files to individual subsystems using the MMS file transfer services. The operator uses a Dexterity screen to enter the file name to be transferred, the destination file name and the destination subsystem.
- 5) Subsystem event or alarm conditions are supported in two ways. Both methods combine the MMS information report service with the Dexterity alarm service. In the first method, the subsystem is programmed to report data based on changing conditions. The subsystem software engineer can report on change, report on critical condition and/or report on return to nominal condition. The Dexterity database is then configured to alarm the operator when a reported data element goes outside its limits. In the second method, the subsystem is programmed to report a text message to Station Monitor and Control. The conditions that warrant a text message are left to the discretion of the subsystem software engineer. The text message is sent to Station Monitor and Control using the MMS information report service. The Dexterity database is configured to display these text messages as events or alarms as a function of the message type.

- 6) Monitor data is exchanged between directly between subsystems based on negotiated interface agreements using the MMS read variable service or the MMS information report service.

Subsystem Integration

All of the subsystems at the Venus site are based on the Intel 80x86 computer and the DOS operating system. Personal computers and the DOS operating system are frequently used by experimental subsystems because they are inexpensive and simple to program. DOS is a "single user" operating system and supports execution of only one program at a time. The subsystem program is structured to establish the subsystem configuration and status on execution, and to run in a continuous loop until an operator interruption triggers termination.

The key element in the application of MMS is the abstract model called the Virtual Manufacturing Device (VMD). The VMD model is used to describe the externally visible characteristics of a real device. Software modules which manipulate a real device are called VMD objects. These objects can be manipulated using MMS services such as context management, variable access, domain management, semaphore management, and event management services. State changes detected in the real device and defined in the VMD model can trigger MMS services. For the implementation at Venus, a minimum of effort was required to make the subsystems appear as VMDs.

To minimize the impact of the monitor and control effort on subsystem development, software routines were developed to provide network integration with a minimum of impact to the subsystem code. The impact to subsystem code is reflected in the pseudo-

code below:

```
main()
{
    dsn_init_mms(); /* routine called to set up the
                       network environment */
    set_up_subsystem_environment();
    while( not_interrupted_by_operator )
    {
        monitor_subsystem_operations();
        execute_subsystem_commands();
        dsn_comm_server(); /* routine called to check
                              for network messages */
    }
    set_subsystem_in_safe_shut_down_conditions();
    dsn_terminate_mms(); /* routine called to terminate
                           the network environment */
    exit();
}
```

A summary of the MMS services and routines developed for the subsystems and the Station Monitor and Control workstation is provided below.

MMS Context Management Services

The routine *dsn_init_chan()* is used by a client subsystem to initiate communications with a server.

The routine *dsn_conclude()* is used by a client subsystem to close an association with a server subsystem.

The routine *dsn_listen()* is used by a server subsystem to open a

communication resource. The communication resources for each subsystem, commonly referred to as channels are established when the subsystem initiates operation. A subsystem that operates as a server will allocate listen channels for clients establish communications. The *dsn_listen()* routine is provided to permit a subsystem to allocate additional channels as server channels while operating. The *dsn_stop_listen()* is used to close a channel in the listen mode.

The routine *dsn_abort()* is used by a client to terminate a client-server association without regard for pending requests. This service is intended for use under special conditions and not to be used in place of the *dsn_conclude()* service.

MMS File Management Services

The routine *dsn_copy_file()* is used by a client subsystem to copy a file from a server subsystem.

The routine *dsn_send_file()* is used by a client subsystem to invoke the MMS obtain file service in which a server subsystem copies a file from a client subsystem.

The routine *dsn_delete_file()* is used by a client subsystem to delete a file from a server subsystem.

MMS Variable Services

The routine *create_dsn_mms_types()* is used to establish a variable as a network object. Subsystem implementors use this routine to register variables that will be managed across the network. The registration process identifies the address of the variable, its type and size to the subsystem network software.

The routine *reg_write_var_indic_func()* is used to identify a function to be called by the server when a specific named variable is written across the network. For example, a server subsystem could have a physical switch that is set according to the value of a named variable. By design, the switch must be reset to its new condition on change of the named variable. The server subsystem software engineer can write a function that affects the physical change of the switch and use the *reg_write_var_indic_func()* routine to associate the function with the named variable. When the write indication is received for the associated named variable, the function is called.

The routine *reg_write_var_confr_func()* is used to identify a function to be called by the client when a specific named variable is written across the network and the write confirmation is received by the client. Extending the example above, a client could have a sequence of actions to perform that are dependent of the server subsystem switch position. The client subsystem software engineer can write a function that affects action that can only occur after the switch reset has been affected. The client function can be associated with the named variable write confirmation using the *reg_write_var_confr_func()* routine. When the write confirmation is received for the associated named variable, the function is called.

The routine *reg_read_var_indic_func()* is used to identify a function to be called by the server when a specific named variable is read across the network. For example, the status of some element of a server subsystem is reflected by a named variable. The subsystem software engineer can write a function to update the status and associate that function with the specific named variable using the *reg_read_var_indic_func()* routine. When the read indication is received, the function will be called and the read response will reflect the latest status.

The routine `reg_read_var_confr_func()` is used to identify a function to be called by the client when a specific named variable is read across the network and the confirmation is received by the client. Extending the example above, the client subsystem software engineer will write a function to take action on the status returned by the server subsystem. That function is associated with the specific named variable using the `reg_read_var_confr_func()` routine. When the read confirmation is received by the client, the function is called.

The routine `reg_info_report_func()` is used to identify a function to be called by the client when a specific named variable is reported by a server using the `dsn_send_info_rpt()` service.

Summary

The Station Monitor and Control system has been operating on a daily basis at the Venus site since December 1992. The operation personnel have developed an extensive knowledge and understanding of the system. The station personnel perform routine configuration management and develop new tools and displays as conditions change.

The implementation of the Venus Station Monitor and Control has demonstrated that the MMS/OSI protocols can be used to support interprocessor communications in a spacecraft tracking station. The integration of a commercial user interface package with an MMS network server has created a powerful, flexible tool to support the constantly changing requirements at the site. Most of the cost savings derived from the MMS/OSI approach comes from the application of commercial products. The software developed in this activity to support the MMS network services on Station Monitor and Control is less than 20,000 lines of code. The

commercial software products represent more than 750,000 lines of code. The functional capabilities provided through the combination of these two technologies offer great potential for cost savings.

The integration of the two commercial products required three months with an additional three months of comprehensive testing before delivery to the station. We have recently integrated a different SCADA product with the same MMS product on a Sun SPARC. Though the API for the new SCADA product was different, the process only required two months. The table driven approach to the network interfaces and the SCADA products provide a flexible environment for station personnel to respond to changing conditions.

The MMS/OSI communications architecture also provides a flexible environment to changing requirements. The Virtual Manufacturing Device model for subsystems provides a single interface to network services. When the High Resolution Microwave Survey project discovered problems with their interface to the antenna equipment, the implementation of a network gateway was a straight forward process, requiring no modifications to the network.

The most difficult part of the effort was upgrading the individual subsystems to enable them to operate over the LAN. Because the decision to proceed with an integrated, centralized M&C system was made after the subsystems began development, the subsystems were, in essence, retrofitted to support the new architecture. Fifty percent of the Station Monitor & Control effort was spent modifying the subsystems to include a LAN interface, addressing problems which occurred due to the limitations of the DOS operating system chosen by the subsystems, and integrating a variety of commercial packages to support

subsystem-specific problems.

The Venus Architecture is now serving as a baseline for testing automation concepts. The open, flexible architecture has greatly reduced integration time. Venus is serving as an operational testbed for the Link Monitor & Control Operator Assistant, an Artificial Intelligence-based tool which automates configuration, calibration, test, and operation of ground station equipment [3].

Acknowledgments

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

- [1] W. Randy Heuser, "An OSI Architecture for the Deep Space Network," TDA Progress Report 42-109, Jet Propulsion Laboratory, Pasadena, CA, March 30, 1992.
- [2] W. Randy Heuser and Lynne P. Cooper, "An OSI Architecture for the DSN," SpaceOps '92, Pasadena, CA, November 1992.
- [3] Lorraine F. Lee and Lynne P. Cooper, "Operations Automation Using the Link Monitor & Control Operator Assistant," to be published in the Proceedings of AIAA Computing in Aerospace '9, San Diego, CA, October 1992

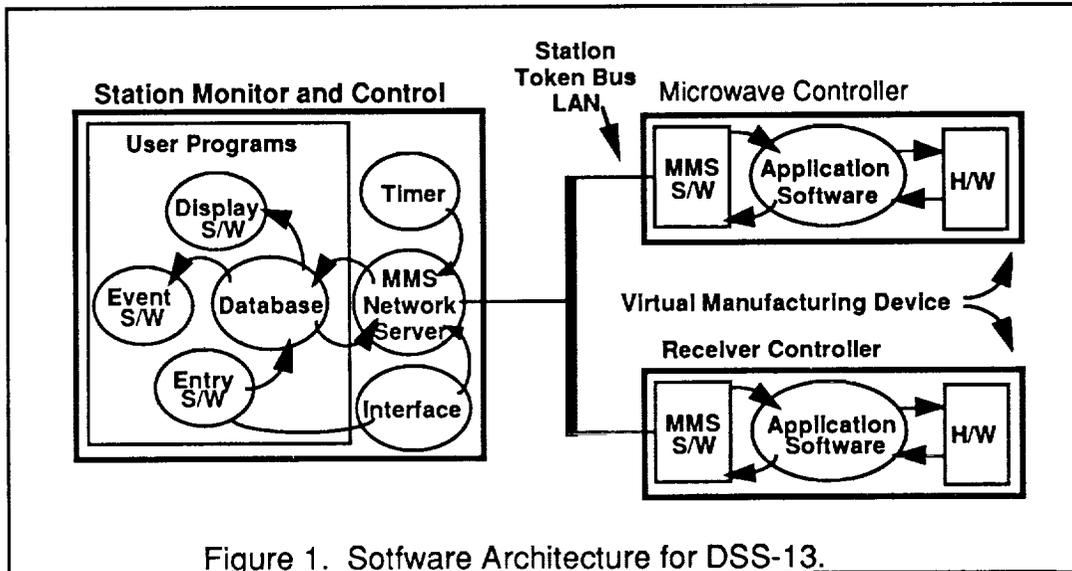


Figure 1. Software Architecture for DSS-13.

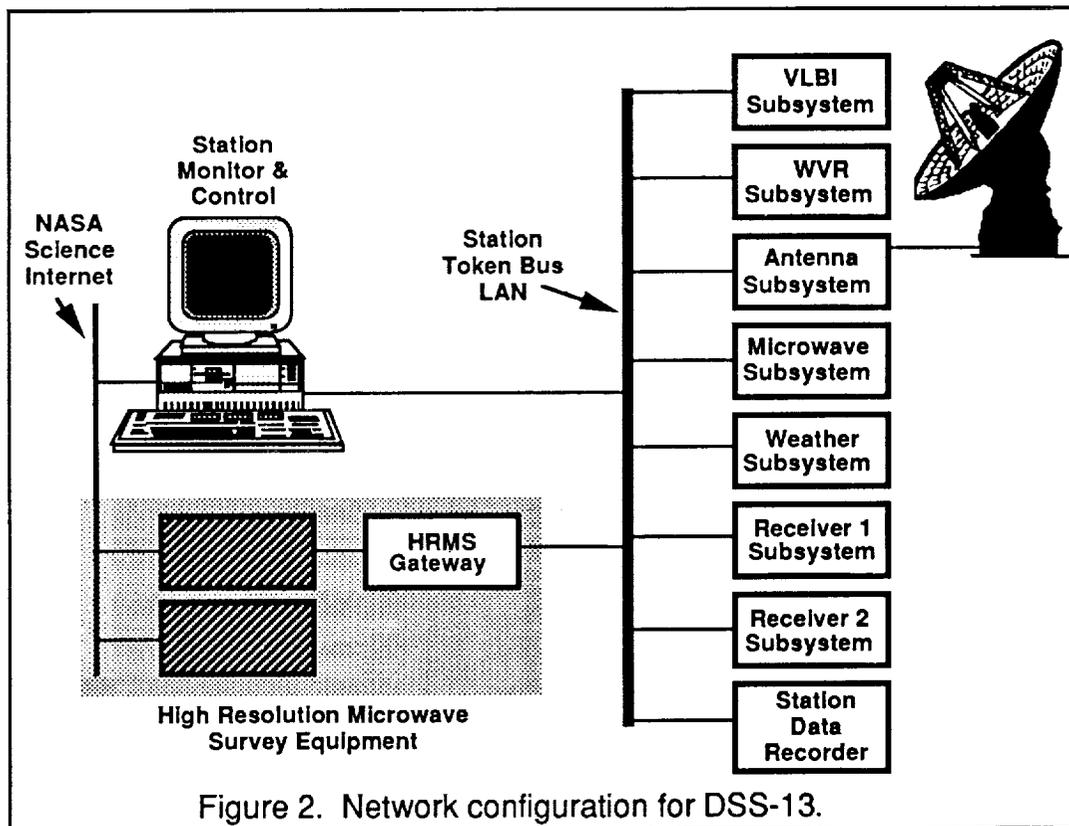


Figure 2. Network configuration for DSS-13.



SESSION 3: NETWORK CONTROL,
SCHEDULING, AND MONITORING



PREDICTING SCHEDULING SUCCESS

Dr. Fredric Messing
Computer Sciences Corporation

Summary

This paper provides an analytical formulation to predict scheduling success for a class of problems frequently referred to as activity scheduling. Space Network communications scheduling is an example of activity scheduling. The principal assumption is that the activity start times are randomly distributed over the available time in the time line.

The formulation makes it possible to estimate how much of the demand can be scheduled as a function of the demand, number of resources, activity duration, and activity flexibility. The paper includes computed results for a variety of resource and demand conditions. The results demonstrate that even with highly flexible activities, it is difficult to schedule demand greater than 60 percent of resources without the use of optimization and conflict resolution capabilities in the scheduling system.

PREDICTING SCHEDULING SUCCESS

Dr. Fredric Messing
Computer Sciences Corporation

Abstract

An analytical formulation is derived to predict the success of scheduling activities on discrete multiple resource time lines using sequential approaches. Success is defined in terms of the probability of scheduling a single activity and the number and cumulative duration of scheduled activities. The results are extended to include scheduling activities with flexible start times. The principal assumption is that the activity start times are randomly distributed over the available time in the time line.

Introduction

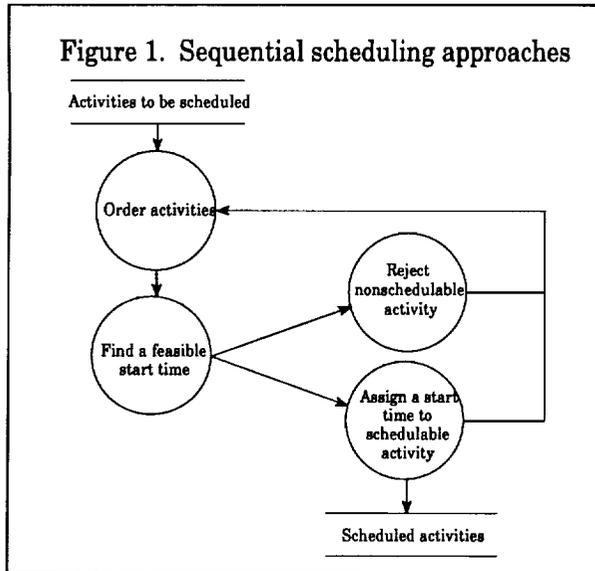
This analysis addresses a type of scheduling problem frequently referred to as activity scheduling. Each activity is assumed to use one of a set of equivalent resources. Each resource can be used to perform only one activity at any time. The activities are independent and have no predecessor relationships. Each activity has a specified duration and start time, although the possibility that the start time is flexible is also considered.

Scheduling becomes challenging when not all

of the activities can be scheduled because of conflicting demand for the resources. The question then becomes: which activities get scheduled and at what times, and which do not get scheduled?

Ideally, an objective should be defined that can be used to identify the optimal schedule and select a scheduling approach that achieves or nearly achieves an optimal schedule as defined by that objective. A typical objective might consist of maximizing the sum of values assigned to each scheduled activity. If the value were the same for each activity, then maximizing the value is equivalent to maximizing the number of scheduled activities. If the value were proportional to the duration of the activity, then maximizing the value is equivalent to maximizing the total time scheduled.

Because optimally solving such problems is complex, most approaches do not attempt to achieve optimality directly and have resorted to a sequential scheduling approach (see Figure 1). A sequential scheduling approach typically begins by using heuristically determined metrics to order the activities by priority. It then considers each activity in



priority order and attempts to find an available time for using one of the resources. Assuming such a time exists, a second heuristic approach determines the start time. If previously scheduled activities conflict with all possible start times, the activity is not scheduled. In either case, the next activity on the list is then considered for scheduling.

The principal method for evaluating scheduling approaches is to determine the extent to which the objective is met. Evaluation is typically accomplished by establishing benchmark problems and generating test schedules. The evaluation criteria generally include the fraction of activities and the fraction of activity time that gets scheduled. This paper provides an analytical technique for predicting scheduling success in these terms.

A closely related issue is the fraction of available resource time that gets scheduled. Providing resources is generally costly, and, before spending money to provide additional resources, managers want to be sure that the existing resources are being used efficiently to perform the specified activities.

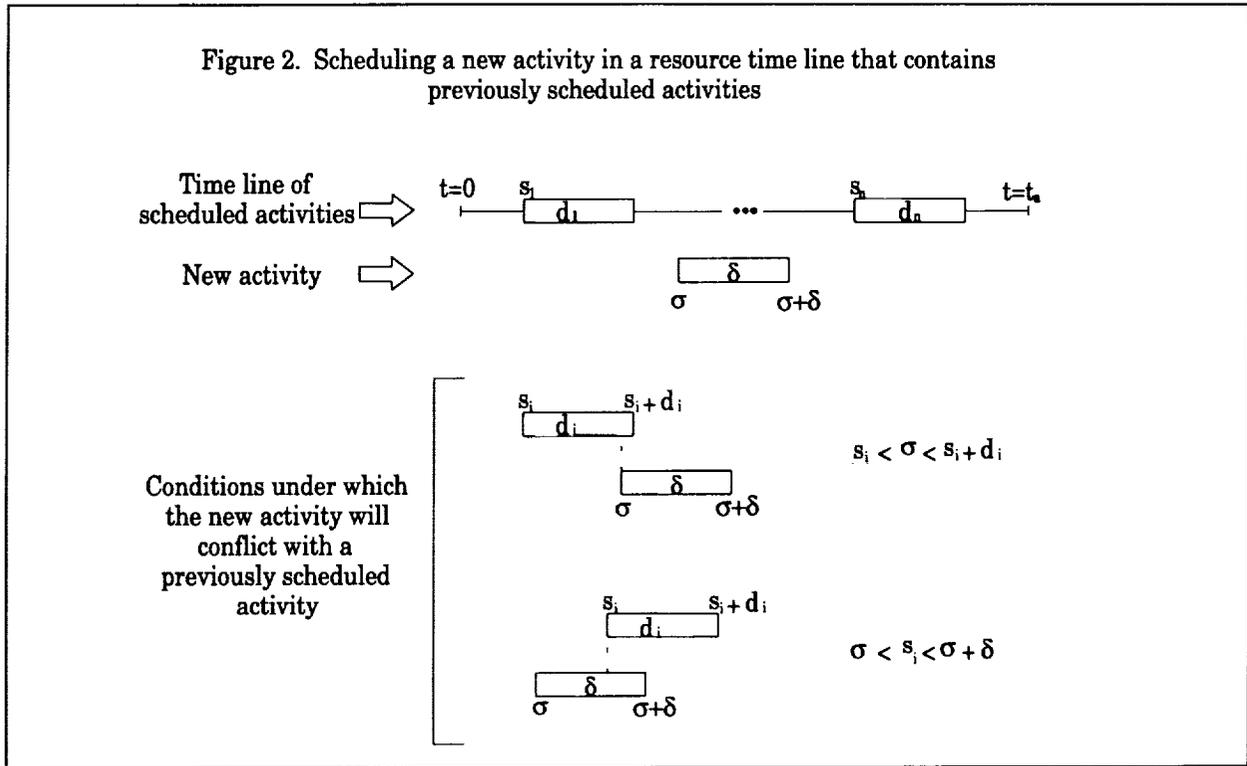
The first part of this paper presents the derivation of the probability of a single additional activity being successfully scheduled on either a single or multiple resources. An iterative process is then defined to determine the overall probability of successfully scheduling any number of activities. Next the benefits to improved scheduling success from start-time flexibility are included. Finally, the distribution of gaps remaining in the time line is discussed.

Single-Activity Scheduling Success

Scheduling success probability is derived by considering an attempt to schedule a single additional activity on a resource time line that contains a number of activities already scheduled. Given a single, discrete, resource time line (see Figure 2) with n randomly scheduled activities at start times and with durations

$$\{s_i, d_i\}_{i=1,n}$$

Figure 2. Scheduling a new activity in a resource time line that contains previously scheduled activities



such that

$$s_i + d_i \leq s_{i+1}$$

consider a new activity with duration δ and random start time σ . The new activity will not be scheduled successfully if its start time conflicts with any previously scheduled activity ($s_i \leq \sigma \leq s_i + d_i$) or if a previously scheduled activity has a start time that conflicts with the new activity ($\sigma \leq s_i \leq \sigma + \delta$).

Since σ is uncorrelated with any previously scheduled activity, the probability that it will not conflict with a previously scheduled activity is given by the fraction of the time

line remaining unscheduled

$$\left(\frac{t_r}{t_a} \right)$$

where

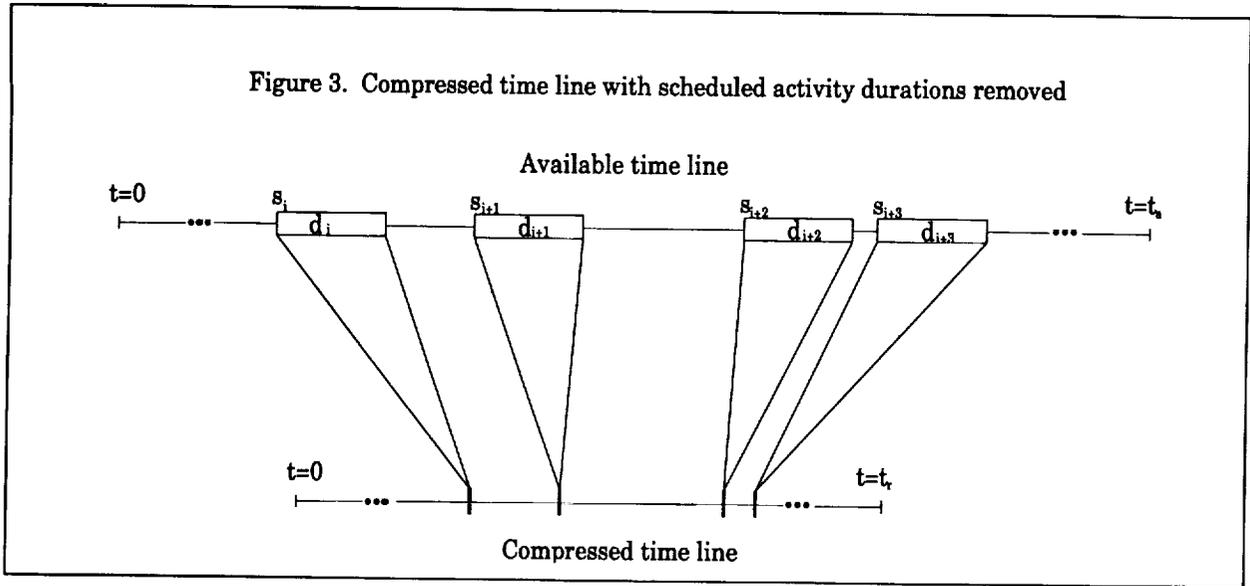
t_a = the length of the time line

and the remaining time is given by

$$t_r = t_a - \sum_{i=1,n} d_i$$

The probability that no previously scheduled activity has a start time that conflicts with the new activity is determined by considering a compressed time line of length t_r (see Figure 3), generated by removing the scheduled activity durations from the

Figure 3. Compressed time line with scheduled activity durations removed



available time line. The previously scheduled activities appear with zero duration randomly distributed throughout the time line. For each of the n previously scheduled activities, the probability that start time s_i will not conflict with the new activity ($s_i < \sigma$ or $s_i > \sigma + \delta$) is given by

$$\left(1 - \frac{\delta}{t_r}\right)$$

under the assumption that

$$\delta \ll t_r \text{ and } d_i \ll t_a$$

to avoid any effects from the ends of the time line. Combining the probability that σ is not in conflict with any previously scheduled activity, with the probability that none of the s_i are in conflict with the new activity, results in the probability P_1 that the new activity can be scheduled

$$P_1 = \left(\frac{t_r}{t_a}\right) \left(1 - \frac{\delta}{t_r}\right)^n$$

In the limit that n becomes large while (t_r/t_a) remains fixed

$$P_1 \rightarrow \lim_{n \rightarrow \infty} P_1 = (1 - L) e^{-\left[\frac{L}{(1-L)} \langle d \rangle\right]}$$

where

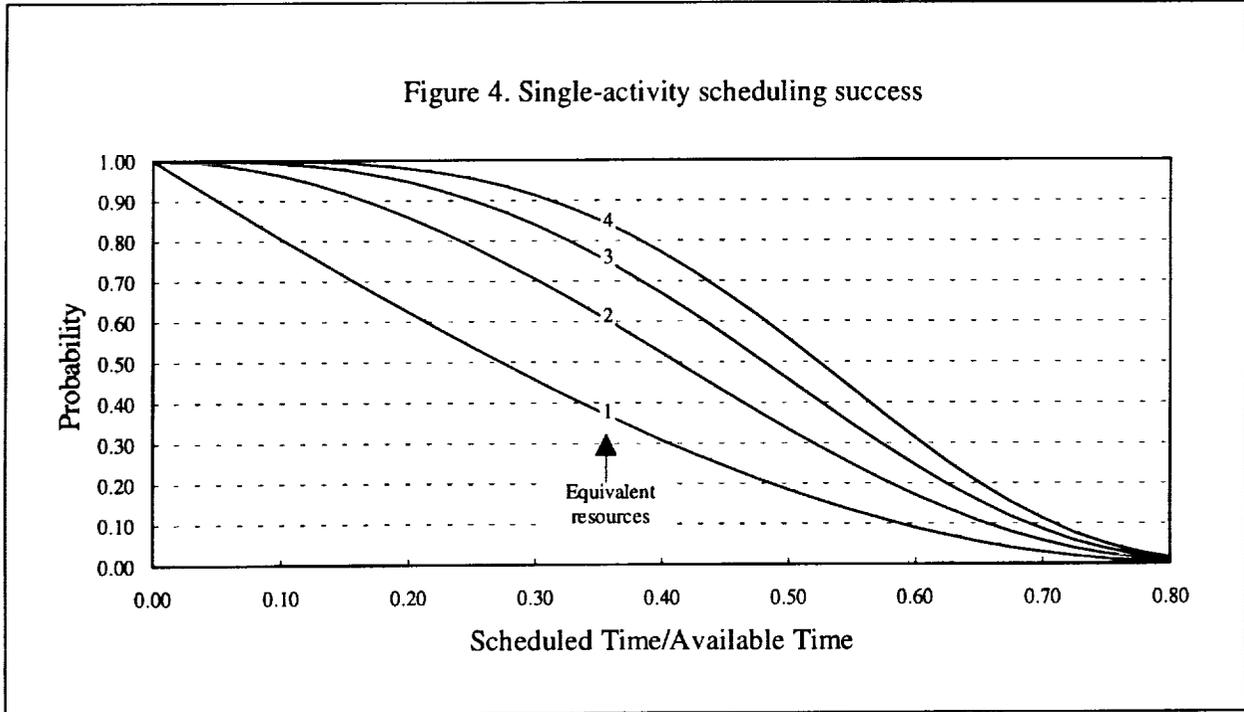
$$L = \frac{\sum_{i=1,n} d_i}{t_a} = 1 - \frac{t_r}{t_a} = \text{scheduled load}$$

$$\langle d \rangle = \frac{1}{n} \sum_{i=1,n} d_i = \frac{L t_a}{n}$$

= average duration of scheduled activities

If each activity can be scheduled on any of m equivalent unconstrained resources, then P_1

Figure 4. Single-activity scheduling success



becomes the probability that the new activity can be scheduled on each of the m resources. The probability of successfully scheduling an additional activity on any of the m resources is then

$$P_m = 1 - (1 - P_1)^m$$

Figure 4 shows the probability of successfully scheduling a new activity of average duration as a function of the already scheduled load. For small values of L , the probability decreases as $1 - L^m$. For larger values of L , the exponential causes the probability to fall more rapidly. For a single resource, by the time L has reached 30 percent, the scheduling success for a new activity has fallen to 46 percent. Four equivalent resources are required to keep the single-activity scheduling success above

50 percent for 50 percent loading.

Integrated Scheduling Success

Scheduling success integrated over all activities is determined by applying P_m iteratively. Consider N activities with random start times to be scheduled on m equivalent resources. The first m activities can each be scheduled successfully without conflict, one activity on each resource. Therefore, the number of activities and the scheduled load are

$$n(m) = m$$

$$L(m) = \frac{\sum_{j=1, m} d_j}{t_a}$$

For activities $j=m+1, N$, the steps are

- 1: Compute the probability $P_m(j)$ to schedule activity j , given $n(j-1)$ previously scheduled activities out of $j-1$ attempts
- 2: Update the number of activities scheduled out of j attempts $n(j) = n(j-1) + P_m(j)$
- 3: Compute the scheduled load $L(j) = L(j-1) + \frac{P_m(j)d_j}{t_a}$

The average scheduling success can then be computed as the ratio of the number of scheduled activities to the number of attempted activities

$$\frac{n(N)}{N}$$

or as the ratio of the scheduled time to the attempted time

$$\frac{t_a L(N)}{\sum_{j=1, N} d_j}$$

If all of the activities are of the same duration, then the two measures are identical. Figure 5 illustrates the integrated scheduling success when all activities are of the same duration. If the demand is for 50 percent of the available time on 4 resources, 90 percent of the activities will be scheduled successfully. With 2 resources, 79 percent will be scheduled successfully. If the demand is for 70 percent of the resources, the success for 4 resources drops to 79 percent and it drops to 69 percent for 2 resources.

Start-Time Flexibility

The activity start times for some scheduling

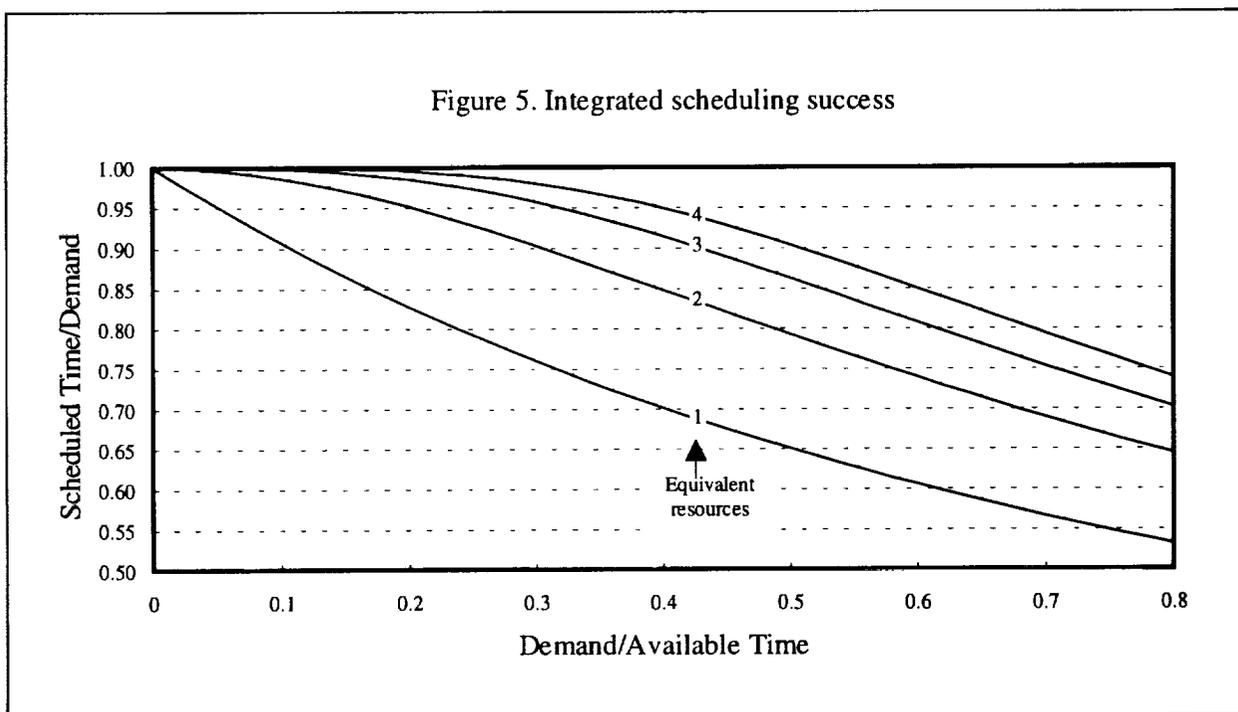
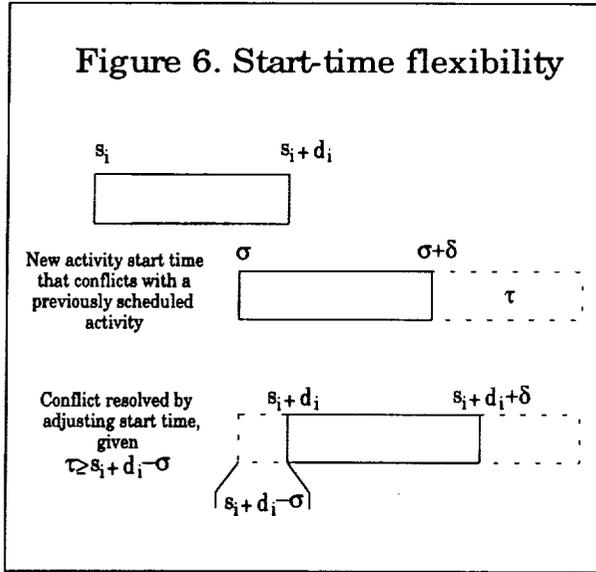


Figure 6. Start-time flexibility



problems are not fixed. Rather, they have flexibility τ such that they can be scheduled to start at any time between σ and $\sigma + \tau$. The benefit of the start-time flexibility can be determined by considering an activity (see Figure 6) with start time σ that conflicts with a previously scheduled activity of duration d_i , scheduled to start at time s_i , $s_i \leq \sigma \leq s_i + d_i$. If $\tau \geq s_i + d_i - \sigma$, then the conflict with activity i can be resolved by adjusting the new activity start time to $s_i + d_i$. The probability that this resolution is possible is given by

$$\left(\frac{\tau}{d_i} \right)$$

for

$$\{\tau < d_i\}$$

For larger values of τ , the probability of resolving the conflict is 100 percent. The

average probability for resolving a conflict with the new event start time is given by the product of the probability that the new activity conflicts with the previously scheduled activity i , multiplied by the probability that a conflict with activity i can be resolved, summed over all previously scheduled activities

$$\sum_{i=1,n} \left(\frac{d_i}{t_a} \right) \left(\frac{\tau}{d_i} \right) = \frac{n\tau}{t_a} = L \frac{\tau}{\langle d \rangle} = Lf$$

where f is the normalized flexibility

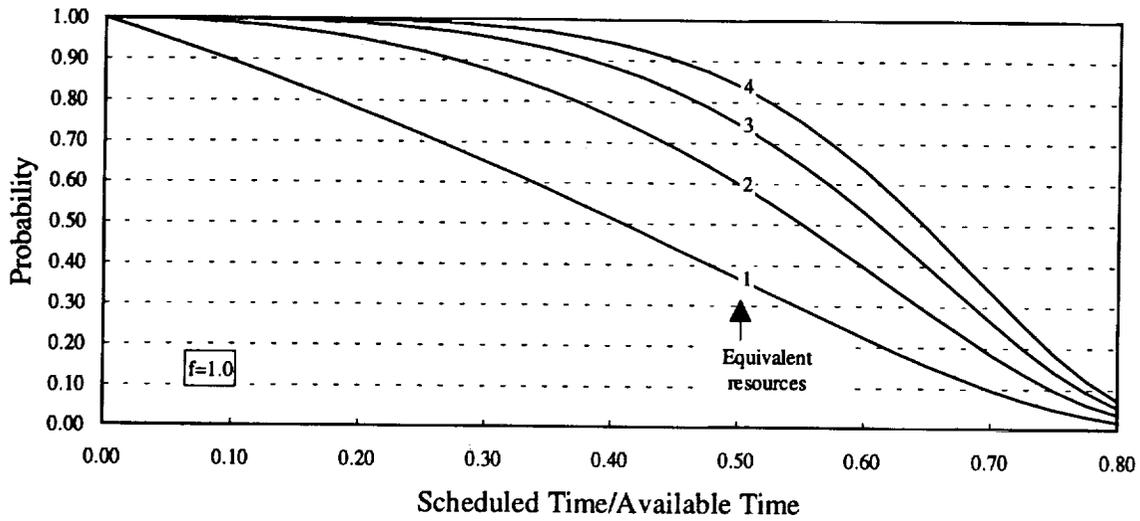
$$f = \frac{\tau}{\langle d \rangle}$$

The probability of successfully scheduling the new activity is determined by multiplying Lf by the probability that another activity will have been scheduled in conflict with the adjusted activity and by adding the result to the previously determined value of P_1

$$P_1 = (1 - L + Lf) e^{-\left[\frac{L}{(1-L)} \frac{\delta}{\langle d \rangle} \right]}$$

Figure 7 shows the probability for successfully scheduling a new activity of flexibility $f = 1$ of average duration as a function of the already scheduled load. Because of the flexibility, this data shows significant improvement in scheduling success when compared with Figure 4. For a single resource scheduled at 30 percent of available

Figure 7. Single-activity scheduling success with flexible start times

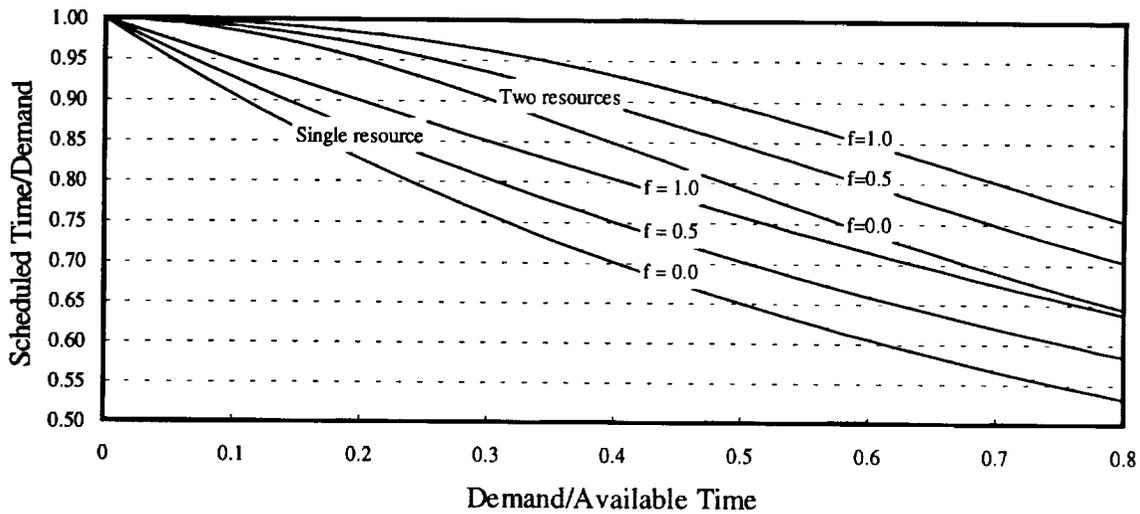


time, the scheduling success for a new activity increases from 46 percent to 65 percent. With 4 equivalent resources, 50 percent scheduling success can be maintained up to a demand of

65 percent of available resources.

The improvement in integrated scheduling success is illustrated in Figure 8. Increasing flexibility from $f = 0$ to $f = 0.5$ increases the

Figure 8. Integrated scheduling success with varying flexibility



scheduling success by approximately 5 percent for high levels of demand. Increasing flexibility to $f = 1$ increases scheduling success by an additional 5 percent.

Duration Flexibility

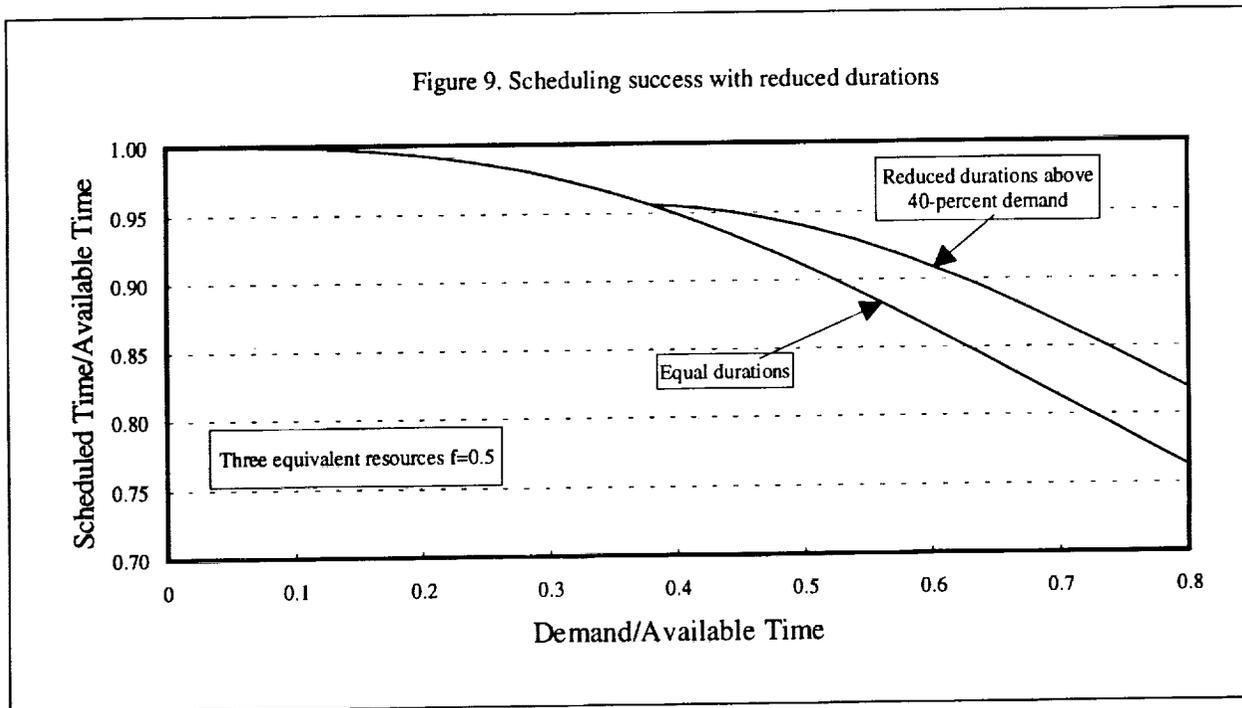
As previously indicated, the exponential term in P_1 dominates the linear term for larger values of scheduled load L . The impact of the exponential term can be partially reduced by first scheduling the larger duration activities and then scheduling the shorter duration activities.

Figure 9 illustrates scheduling success when the demand above 40 percent is divided into twice the number of activities, each activity of half the duration of the activities below 40

percent. This success is compared to scheduling success when all activities have the same duration. With the reduced-duration activities, scheduling success is increased by approximately 5 percent. In practical applications, durations of unschedulable activities can be reduced to improve scheduling success.

Highly Flexible Start Times

As the flexibility of start times increases, the probability of successfully scheduling an activity increases. For values of $\tau > d_i$, conflicts of the new activity start time σ with a previously scheduled activity can always be resolved (see Figure 6) by delaying the new activity to start at the end of the conflicting



activity. The linear term in the scheduling probability is eliminated, and the single resource success probability becomes

$$P_1 = e^{-\left[\frac{L}{(1-L)} \frac{\delta}{\langle d \rangle}\right]}$$

This probability is actually a lower bound. The new activity is also schedulable if the start time s_i of a previously scheduled activity conflicts with the new activity and if $\sigma + \tau > s_i + d_i$.

For values of $\tau > d_i + d_{i+1} + \delta$ (for an average-duration activity $\delta = \langle d \rangle$, this value corresponds approximately to $f > 3$); the success probability increases further, as illustrated in Figure 10. If the length g_i of the first gap following the start time σ of the new activity is less than the duration of the new activity, $g_i < \delta$, then the new activity will not

be schedulable in that gap. This probability is given by

$$\left(1 - e^{-\left[\frac{L}{(1-L)} \frac{\delta}{\langle d \rangle}\right]}\right)$$

If the new activity is not schedulable in the first gap, then its start time can be delayed to the end of the next previously scheduled activity. The probability is identical that the gap following this activity is also too small in which to schedule the new activity. Consequently, the probability that the new activity will be schedulable in one of these two gaps is given by

$$P_1 = 1 - \left(1 - e^{-\left[\frac{L}{(1-L)} \frac{\delta}{\langle d \rangle}\right]}\right)^2$$

As τ increases further, scheduling success continues to increase. Figure 11 illustrates the

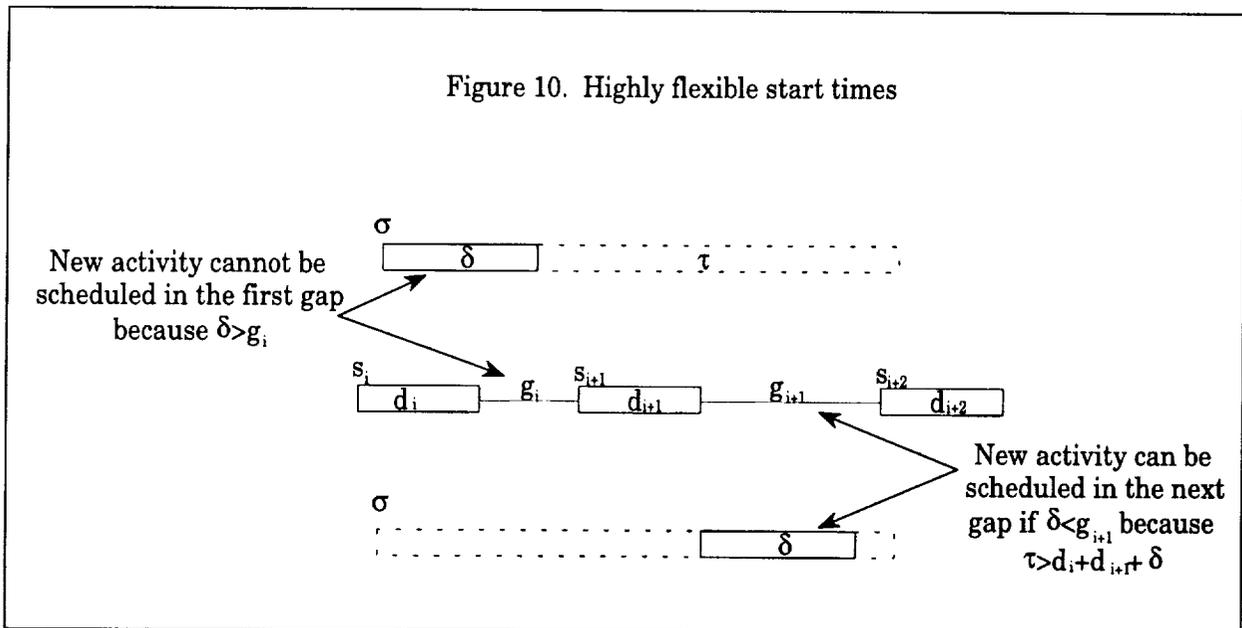
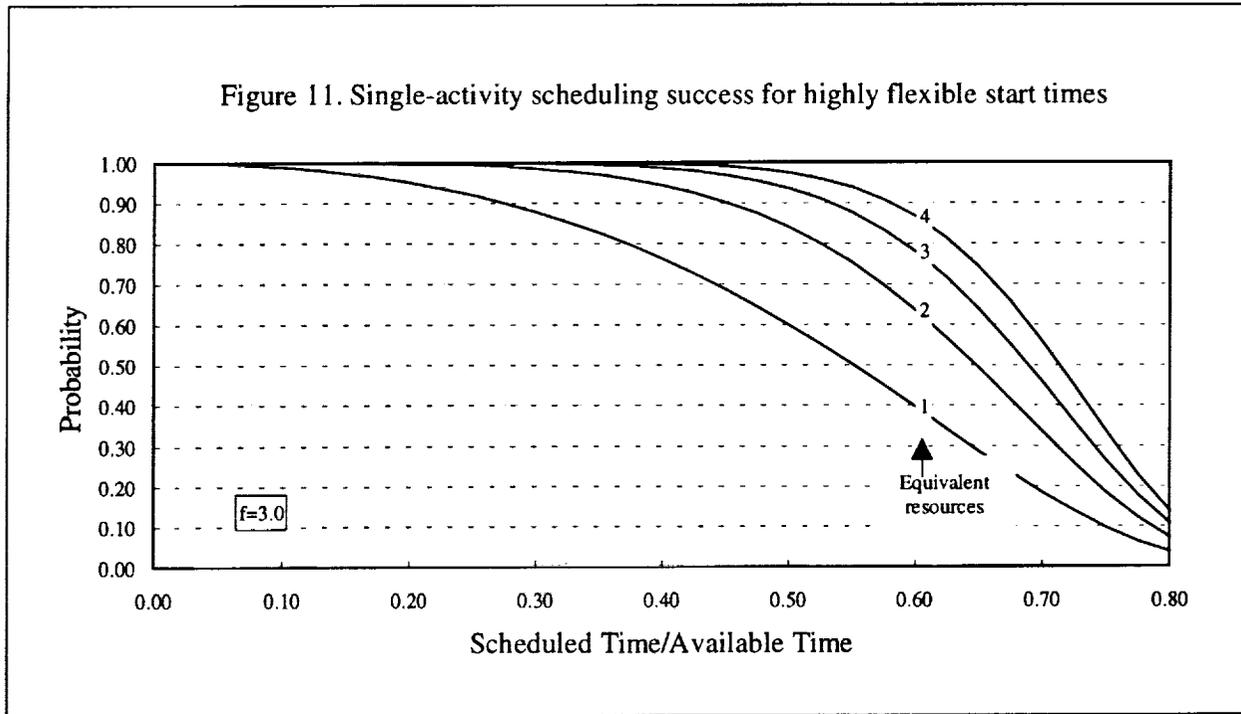


Figure 11. Single-activity scheduling success for highly flexible start times



significant gain in scheduling success that accrues from this added flexibility for an average-duration activity. It also illustrates that, even with this amount of flexibility, it is difficult to successfully schedule beyond a demand of 60 percent of the resources.

This conclusion depends on the assumption that scheduled activities are placed randomly within their schedulable start-time flexibility. Techniques exist for selecting start times to optimize resource use. These techniques effectively reduce the size of small gaps and increase the size of large gaps, thereby improving scheduling success.

Gap Size Distribution

The difficulty in scheduling more than 60

percent of the resources results from both the time line being heavily scheduled and the remaining gaps being too small for additional activities to be scheduled. This problem can be understood by determining the distribution of gap sizes.

The probability $v(g)$ for an individual gap to have a length greater than value g can be determined by selecting the end, $s_i + d_i$, of any scheduled activity and by considering the probability that no other activity is scheduled within gap g from this point. This probability is precisely the same probability derived earlier for scheduling an activity with a nonconflicting start time

$$v(g) = e^{-\left[\frac{L}{(1-L)} \frac{g}{d}\right]}$$

The distribution of gap sizes (see Figure 12) is given by the probability of finding a gap between g and $g+dg$

$$-\langle d \rangle \frac{dv(g)}{dg} = \frac{L}{(1-L)} e^{-\left[\frac{L}{(1-L)} \frac{g}{\langle d \rangle} \right]}$$

As the resource becomes more heavily scheduled, the remaining gaps become significantly smaller than the average activity duration, making them unusable for scheduling average-duration activities. The amount of time $T(g)$ remaining in gaps larger than g is given by

$$\begin{aligned} T(g) &= \int_g^{\infty} n g \frac{dv(g)}{dg} dg \\ &= n \left(g + \frac{(1-L)\langle d \rangle}{L} \right) e^{-\left[\frac{L}{(1-L)} \frac{g}{\langle d \rangle} \right]} \end{aligned}$$

The ratios of $T(g)$ to t_a and t_r are given by

$$\frac{T(g)}{t_a} = \left(\frac{Lg}{\langle d \rangle} + (1-L) \right) e^{-\left[\frac{L}{(1-L)} \frac{g}{\langle d \rangle} \right]}$$

and

$$\frac{T(g)}{t_r} = \left(1 + \frac{Lg}{(1-L)\langle d \rangle} \right) e^{-\left[\frac{L}{(1-L)} \frac{g}{\langle d \rangle} \right]}$$

which, in the case $g = \langle d \rangle$, go to

$$\frac{T(g)}{t_a} = e^{-\frac{L}{(1-L)}}$$

$$\frac{T(g)}{t_r} = \frac{1}{(1-L)} e^{-\frac{L}{(1-L)}}$$

Figure 13 illustrates the fraction of the time line remaining in gaps larger than the average activity duration as a function of scheduled load L . For example, when 50 percent of the time line has been scheduled, only 37 percent of it consists of gaps larger than an

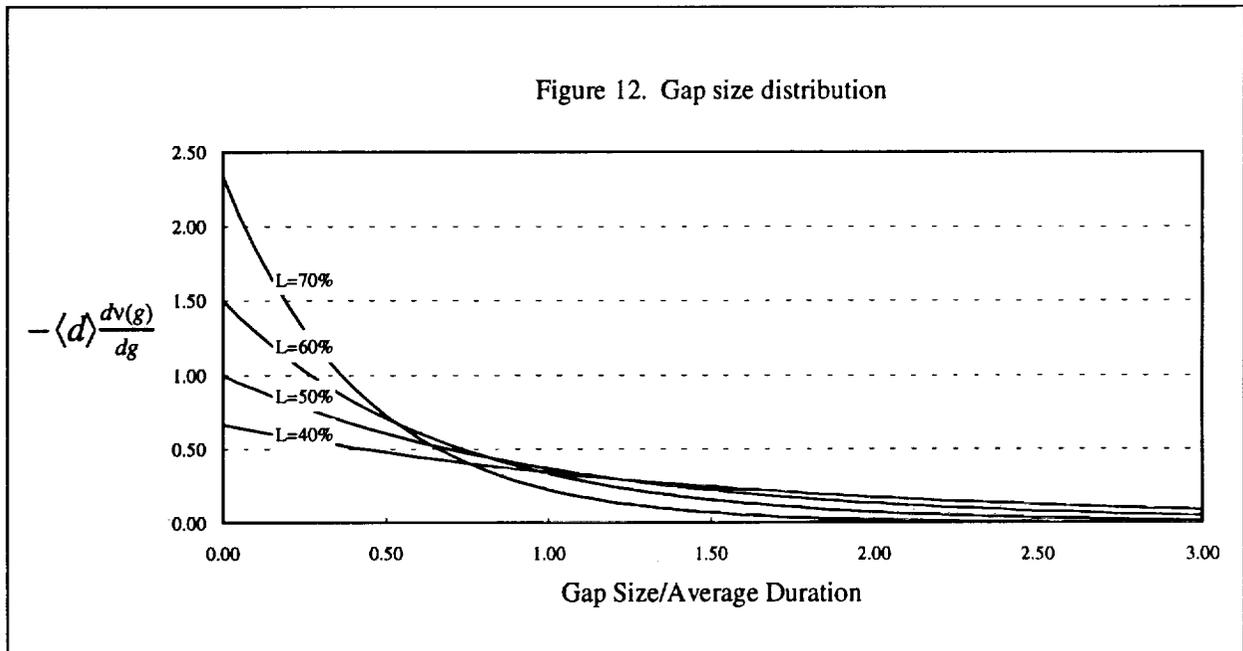
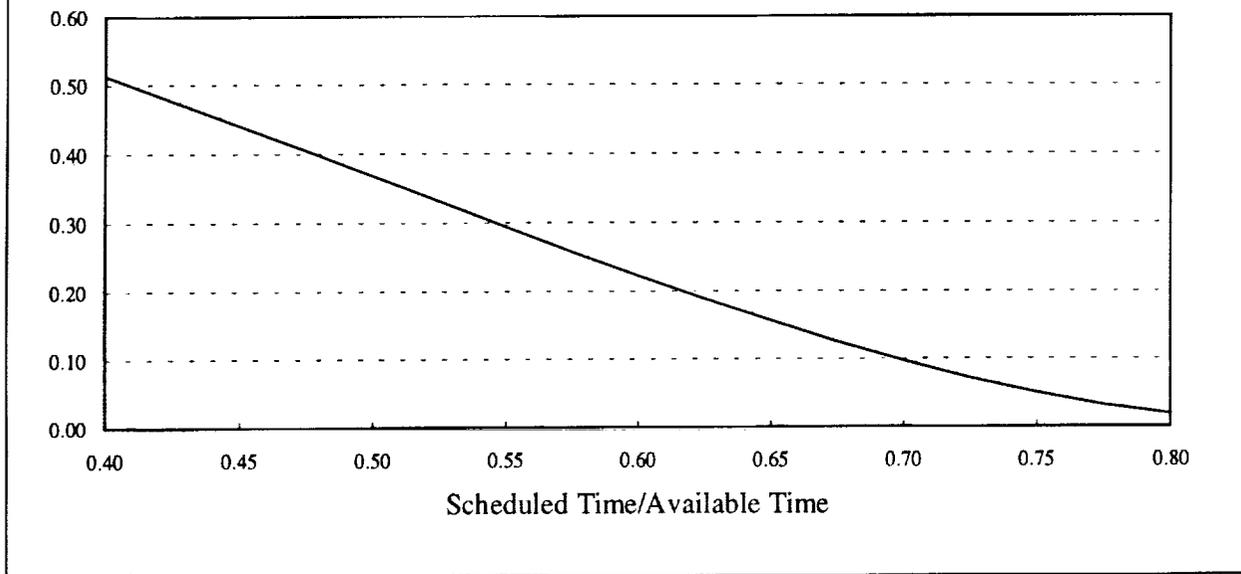


Figure 13. Fraction of time line in gaps larger than average-duration activities



average-duration activity. When L reaches 70 percent, only 10 percent of the time line consists of gaps larger than an average-duration activity. The remaining 20 percent is in gaps that cannot be used to schedule activities of average or larger duration. The time in these gaps can be recovered by adjusting activity start times to reduce the size of small gaps and increase the size of large gaps.

Conclusion

This paper provides formulae to compute success for scheduling individual activities with start-time and duration flexibility on single or multiple resources. An iterative technique is presented for determining

scheduling success integrated over all schedulable activities. It demonstrates the significant increase in scheduling success that can be achieved when scheduling flexible activities. It also provides a distribution of sizes of gaps remaining in the time line and demonstrates the dramatic decrease in time remaining in large gaps as scheduled time increases.

This analytical formulation can be used to calculate realistic estimates of scheduling success without actually developing schedules. Such estimates can be used for capacity planning or predicting scheduling success for varying combinations of activities and resources.

DISTRIBUTED INTELLIGENT CONTROL AND STATUS NETWORKING

**Andre Fortin
Manoj Patel
NASA/Goddard Space Flight Center
Network Control Systems Branch
Greenbelt, MD 20771**

Summary

Over the past two years, the Network Control Systems Branch (Code 532) has been investigating control and status networking technologies. These emerging technologies use distributed processing over a network to accomplish a particular custom task. These networks consist of small intelligent "nodes" that perform simple tasks. Containing simple, inexpensive hardware and software, these nodes can be easily developed and maintained. Once networked, the nodes can perform a complex operation, without a central host. This type of system provides an alternative to more complex control and status systems which require a central computer. This paper will provide some background and discuss some applications of this technology. It will also demonstrate the suitability of one particular technology for the Space Network (SN) and discuss the prototyping activities of Code 532 utilizing this technology.

Introduction

Over the last two years, under a research project, the Network Control Systems Branch (Code 532) at Goddard Space Flight Center has been investigating distributed intelligent control and status networking technologies. These emerging technologies use distributed processing over a network of smart devices or "nodes" to accomplish a particular custom task, like home automation, without a central host. These networks are designed to be simple and inexpensive for an engineer to develop and implement a custom task. This paper will provide some background and discuss some applications of this technology. It will also demonstrate the suitability of one particular technology for the Space Network (SN) and discuss the prototyping activities of Code 532 utilizing this technology.

Background

A continuing trend to convert from centralized to distributed intelligent control and status systems seems to have become widespread. A wide spectrum of systems designed for industrial automation, building controls, automobiles, and consumer products are following this trend. The reasons for this trend are numerous.

Centralized control and status systems consist of remote sensors that provide feedback to a central microcontroller, which in turn sends signals to monitors and actuators. Each centralized control system is unique with its own input/output and processing requirements. It consists of large computational engines that process the inputs and outputs of a whole suite of sensors and equipment and relay the changes and actions to the appropriate targets. These systems are expensive to develop and install, and very difficult to expand and maintain on an ongoing basis.

Intelligent distributed control and status systems consist of nodes with embedded intelligence where each node performs a simple task. Nodes may be devices such as proximity sensors, switches, motion detectors, and relays. While individual nodes in this system perform a simple

task, the entire system performs a complex operation such as automation of a building, factory or a house, without the need for a central host.

Intelligent distributed control systems have a number of significant benefits over centralized control systems. These systems have lower starting costs by using simple and common methods of communicating between all nodes of the system, like using one protocol. The networking nature of these systems facilitates graceful growth by easing expansion and reconfiguration. These systems cut installation costs by sharing common wiring among nodes. Finally, the flexibility of these systems allows many diverse products and applications to interoperate.

Existing networking technologies such as local area networks (LANs) could be used to network these distributed processing systems. However, the Ethernet protocol used to operate a LAN would be an overkill for the requirements of a control and status network. Ethernet is designed to move huge amounts of user documents, data bases, and graphics files among computers, file servers, and printers. A control/status network involves moving relatively short data frames carrying command and status information that trigger actions within devices. Ethernet is much more complex than what is necessary for a control/status network. A control/status scenario could be implemented with a LAN. Such a system, however, would be a large, expensive development effort with complex hardware and software for each node. New technologies have recently been introduced that provide a full package of hardware and software development tools, a microprocessor or microcontroller, I/O interfaces for interfacing to sensors and actuators, a network operating system, and a communications protocol to develop these distributed intelligent control and status networks.

Examples of Uses

Targeted uses of distributed intelligent control and status networks include industrial, home, and office automation. Particularly custom uses are known to exist for cars and mobile homes. These environments have special needs. The nodes must be small, inexpensive, and sometimes use existing wiring. Some of these networking technologies have multiple physical media available to allow for node to node communication, such as existing power lines, RF, and twisted

pair cable.

A good example frequently given by distributed intelligent control and status networking vendors is a system that controls and monitors common equipment found in the home or office. Consider a node in every device, such as the power outlets, the lights, the light switches, the smoke detector, the major appliances, and the security system. Such a system, in the event of a fire, could disable power going to the outlets, flash the overhead lights near emergency exits, turn off major appliances, like a furnace, and perhaps turn on the security system (if there is no fire alarm, like in a home). This would all be accomplished without a central host. The smoke alarm node would detect the smoke and send the appropriate messages to the applicable nodes. All of these nodes would be networked via the AC power.

Another good example is a system to simplify the wiring in a vehicle. In the dash board and steering column of a vehicle, large amount of cables run from the passenger compartment to the engine department to control various devices around the car. The lights (rear brake lights, parking lights, and headlamps) are just one subsystem within a vehicle that needs controlling from the operator. Presently these devices are controlled by many devoted wires running from the steering column/dashboard to the individual lamps. Consider one wire running between everything, the headlamps, the brake lights, the parking lights, and the switches/controls. Each lamp would contain a node that would control it and each switch/control in the passenger compartment would also have a node. These nodes would signal each other based on operator input. The headlamp node would turn on the high beam if it received a signal from a switch node located near the operator. Because of the networking nature of this technology, the parking light node would ignore the message to activate the high beam since this message is not for that node.

From the previous examples, it can be seen that this networking technology provides for simplification of wiring and providing for automation. Other features include simplicity of engineering and maintenance. Nodes, like in the previous example, that are designed to be light nodes can be generic. Specific features of a light node can be determined at installation time and not at manufacturing time. A building engineer or home owner would not have to stock various

light nodes for repair, special features like blinking during a fire alarm (assuming the node is the closest light to an escape exit) can be programmed at installation time. Ease of engineering is also a good trait of these technologies, due to the nature of the simple hardware and processors used in the nodes.

There are several vendors that offer these technologies in which some are very focused on a particular application. General Motors has a system called CAN, targeted for vehicle automation and control. CEBus, a home automation standard, is available free for any manufacturer to design nodes for. The Local Operating Network (LON), designed by Echelon Corporation, is a flexible technology that can be used in many applications and environments. It may be well suited for systems here at NASA. All of these proprietary distributed intelligent control and status systems can provide homes, offices, industrial plants, and large systems with simple methods to meet their control and status needs.

Local Operating Networks (LON)

The current control/status systems in the Space Network are centralized and card-caged. Some of these systems involved a considerable amount of design and development effort and are very difficult to expand and maintain. Additionally, the equipment utilizes a number of different interfaces such as GPIB, MS 1553, RS232, and RS422. The control systems for this scenario have dedicated, point-to-point connections to each equipment. Therefore, interoperability between the equipment is a major problem that demands a large amount of software overhead and processing requirements from the centralized computer system. LON offers a single chip, networking solution to these problems that is inexpensive, modular, interoperable, flexible, and easy to develop. Other control/status networking technologies that were considered were focused on particular applications and did not provide the flexibility demanded by the Space Network systems. This technology has the most potential for successfully replacing or complementing a variety of centralized data processing, data distribution, and fault isolation and monitoring systems within the Space Network.

The technology centers around a single VLSI chip called the Neuron. This inexpensive,

multiprocessor chip (less than \$10) implements a multitasking network operating system, a networking protocol that conforms to the 7 layer OSI model, and a flexible input/output interface through built-in device drivers. Neuron chips provide all the necessary functionality to intelligently process a node's inputs and outputs and relay that information to other nodes over a local network. One Neuron is required for every node on the distributed control and status network. A variety of nodes can be designed each serving different functions. Once these nodes are created and communicating among themselves, a distributed processing control and status network called a LON is created.

The integrated hardware/software development system of LON allows the engineer to easily design and test a small, simple node with custom hardware tied to the Neuron's generic I/O interface and custom application software that defines the functionality of the node on the network. This software is written in a unique, network programming language called Neuron C. This proprietary language is an extended version of ANSI C that provides support for new data types, statements, and function libraries. These enhancements optimize the chip for hardware interfacing to I/O devices and real-time, node-to-node communications on a network.

Neuron C shields the programmer from the hardware details of input/output processing, the protocol details of the physical medium being used, and the details of packet generation, addressing, and transmission. This makes the application software short and simple and spares the programmer from learning a new protocol. The development system allows the engineer to thoroughly debug the node's hardware with this application code through a process of single-stepping through hardware breakpoints. The development system's capability for network management and monitoring allows the engineer to test the nodes with other nodes on a prototype network on this system and manage the network once all the nodes are installed and operational. Thus, LON's integrated development environment makes node design quick and easy.

The protocol in which these Neuron-chip based nodes talk to each other is called LONTALK. This 7-layer protocol has been optimized for movement of relatively short data frames carrying control/status information. It provides support for multiple transmission media which include

twisted pair (which runs from 4.9Kbps to 1.25Mbps), radio frequency (4.9Kbps to 625Kbps), powerline (10Kbps), and even custom media to suit the designer's specific needs. The protocol is actually transparent to the medium used, allowing a variety custom distributed control and status networks to be easily created.

LON technology simplifies the development and maintenance of distributed, intelligent control and status networks by providing an inexpensive communications and control processor, an integrated hardware/software development system, and a control/status networking protocol in one package. Thus, these distributed nodes are designed, developed, tested, and installed with minimum effort and time. Adding a new node to the system is simply a matter of connecting it to the network and making other nodes on the network aware of the new node. Thus, expansion is simplified. The LONTALK protocol solves the problem of interoperability among different nodes. Thus, Local Operating Networks resolves the problems of increased node development time, increased installation costs, difficult expansion, and a lack of interoperability currently experienced in the design of centralized control and status systems.

Prototyping Activities of Code 532

After researching into several distributed intelligent control and status networking technologies, Code 532 decided to evaluate the LON technology. Under a research project, Code 532 procured the necessary hardware and software to evaluate and prototype LON. A technical paper was written that describes in detail what was accomplished in the research and evaluation phase of the LON. Technical information can be found in this paper located in the Code 532 branch library.

After the research and evaluation phase, a prototype system using LON was placed in an operational environment in the Network Control Center (NCC). This system, called the Block Rate Monitor (BRM), consists of a block counter, a gateway, and a host computer. The block counter is a LON node that passively monitors four NASCOM lines. This node counts the number of NASCOM blocks per second and checks for clock presence. The statistical data is

then passed over a LON dedicated twisted pair cable to the gateway. The gateway is also a LON node that converts data sent over the LON to RS-232. This protocol converter allows a workstation to interface to the LON. The workstation receives the statistical data and provides the operator with bargraphs (see figure 1) and stripcharts (see figure 3) to evaluate existing traffic. The operator can also graph logged data for a historical perspective of line traffic (see figure 2). The workstation logs 20 days worth of traffic statistics for four NASCOM lines.

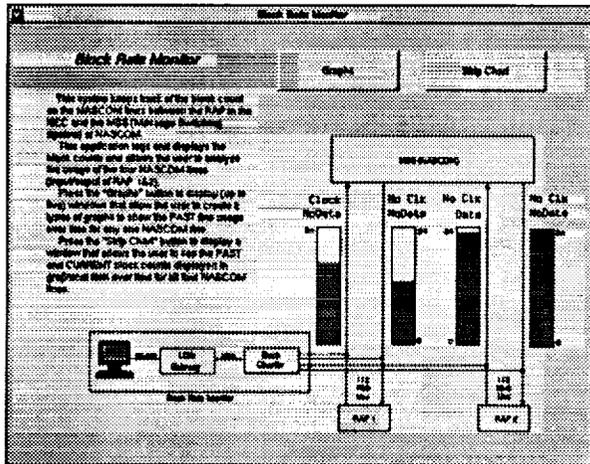


Figure 1. Example main window with bargraphs.

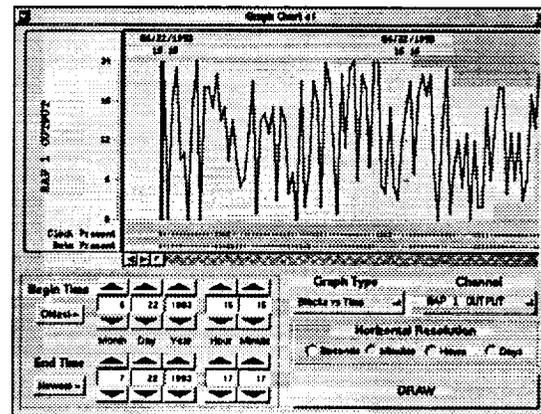


Figure 2. Example of historical data

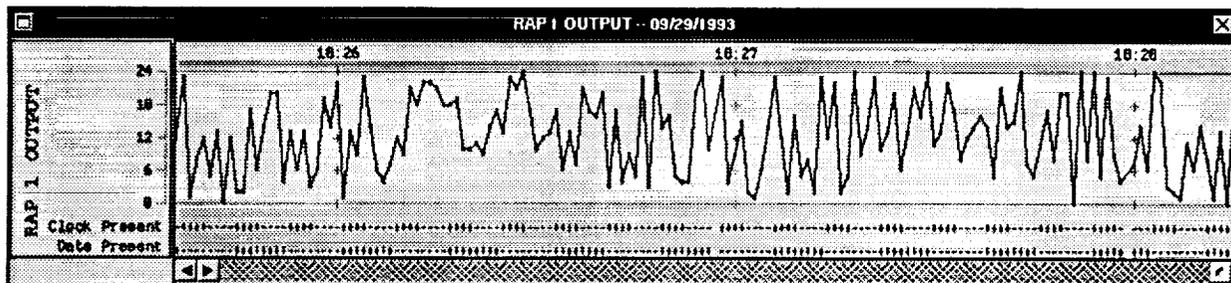


Figure 3. Example strip chart output from BRM application (driven with random data).

With the Block Rate Monitor, the operator can conveniently monitor, in real time, the line activity from a console. The operation of a transmitting system can be deduced as well. If it is not generating any blocks, the system could be down, or there could be a bad cable or patch. This system has already been used to determine if line rate increases were needed (from 112Kb/s to 224Kb/s), but these monitored lines do not have heavy traffic loads and therefore an increase in the line rate was not approved. This system is a very small LON. It only consists of two LON

nodes, a block counter, and a gateway. The next phase of our prototyping a control and status network in an SN environment will greatly expand the LON and the concept of remote traffic monitoring.

The second prototype system is called the Data Integrity Monitoring System (DIMS). This system will consist of 40 LON nodes that each monitor a NASCOM line. Each node, or Data Integrity Monitor (DIM), accumulates statistical data of one NASCOM line and is physically the size of 2 cigarette packs. The cost of each node will be around \$350. These nodes can count blocks, evaluate the CRC, count bad blocks, determine the clock rate, and detect inverted data. With each node monitoring different NASCOM lines, a LON of nodes is created that relays line activity to the operator. Just as in the first prototype, a gateway receives the status information from each DIM and sends it to a workstation.

This workstation will have a new application that will allow the operator to use this system for performance monitoring, fault isolation, and traffic monitoring. With DIMs located on the inputs and outputs of a system, performance monitoring would be achieved. By comparing the graphs of the traffic on the input and the outputs, a time difference should be observed and indicate the time it took for a block to be processed. Since the workstation application has detailed physical diagrams of the site interconnections along with traffic gauges on the lines, fault isolation would be achieved by an operator noting that a line showing no activity should, in fact, have some activity. This would direct the operator to step back in the diagram of the system, to find the break in the system. The operator would have to determine by this application, which patch panel has been mismatched, which cable has gone bad, or which system has failed.

Our prototypes, utilizing the LON technology, are concepts similar to the Small Network Management Protocol (SNMP). SNMP is designed for Ethernet and works with Ethernet host computers. The DIMS, on the other hand, can incorporate any communication line protocol monitor and works as a layer on top of the existing communications lines. The advantages for a DIMS approach include passiveness, non-interference with operational communication lines/systems, and a multi-protocol line/system monitoring. The advantage for an SNMP approach would be a software only system that monitors Ethernet systems on existing Ethernet

host machines. SNMP is more inexpensive and simple, but is limited to Ethernet.

Future plans in these prototyping activities include building nodes that monitor other communication lines (like MIL-STD-1553, GPIB, and Ethernet) and expanding the system to monitor every communication line in the NCC. Future nodes will be much smaller and cost under \$150.

Conclusion

Distributed intelligent control and status networking offers a new and unique solution to the traditional methods of performing control and status. The problems of increased development time, lack of flexibility, lack of interoperability, difficult expansion, and difficult maintenance that we face with centralized control and status vanish with this new method of networked control and status.

Local Operating Networks appears to be on the forefront of this new trend by offering an inexpensive, flexible solution that resolves all of the problems experienced in the design of centralized control and status systems. Our prototyping efforts will determine this technology's feasibility to the Space Network systems.

References

Kleiner, Art. "The Ultimate Control Chip: The Programmable World." Popular Science, May 1992, pp. 90 - 95.

LonWorks Products 1992. USA: Echelon Corporation, 1992.



GENERATING AND EVALUATING ALTERNATE SCHEDULES

Surender D. Reddy

Computer Sciences Corporation, Beltsville, Maryland

Summary

The demand for and availability of Space Network resources are subject to short-term fluctuations and long-term changes. Generation of acceptable schedules under changing demand and resource availability will require the use of different scheduling policies. This paper identifies several such scheduling policies. It defines metrics for evaluating schedules using the criteria directly related to these scheduling policies. Then it applies the metrics to compare several schedules generated for a scenario representative of 1998 SN demand and resources. Finally, the paper describes a method for using these metrics to evaluate schedules based on multiple criteria.

1. Introduction

The demand for and availability of Space Network (SN) resources are subject to short-term fluctuations and long-term changes. Increased demand during shuttle flights is an example of short-term demand fluctuation. Resource unusability because of repair or maintenance of related ground systems is an example of a short-term fluctuation in resource availability. Growth in number of customers and their demand for SN resources is an example of a long-term demand change.

A scheduling policy that results in acceptable schedules under one demand-resource scenario may lead to unacceptable and inappropriate schedules under a different demand-resource scenario. For example, allocation of resources in strict priority order, without regard to other factors, is likely to result in acceptable schedules when the demand to resource availability ratio is low to moderate. The same policy is likely to result in totally unacceptable schedules for lower priority customers when the demand to resource availability ratio is high. Often, which scheduling policy is appropriate for a given demand-resource scenario is unclear. Furthermore, although a scheduling policy generally implies a unique schedule effectiveness criterion, acceptability of a schedule depends on multiple schedule effectiveness criteria.

Therefore, to develop acceptable operational schedules for the use of SN resources, the Network Control Center must:

- Create alternate schedules using alternate scheduling policies
- Evaluate alternate schedules based on multiple schedule effectiveness criteria

This paper identifies alternate scheduling policies and discusses evaluation of the alternate schedules they produce, based on criteria associated with these policies.

2. Alternate Scheduling Policies

Many parameters are associated with the SN resource scheduling problem. Some more important parameters are

- NASA-assigned customer priorities
- Service duration flexibility in terms of nominal and minimum duration
- SN commitments in terms of specified success rates

Potential interdependencies among these parameters imply many possible mutually exclusive scheduling policies:

- Maximize nominal duration events in priority order
- Maximize near-nominal duration events in priority order
- Maximize minimum duration events in priority order
- Satisfy specified event success rates at near-nominal duration in priority order
- Satisfy specified event success rates at nominal duration in priority order
- Satisfy specified time scheduled success rates at nominal duration in priority order
- Satisfy specified time scheduled success rates at near-nominal duration in priority order
- Balance event success rates at near-nominal durations
- Balance event success rates at nominal durations
- Balance schedule time success rates

A scheduling algorithm used for generating schedules should be consistent with the applicable scheduling policies. In addition, even when scheduling policy specific algorithms are available, multiple schedules based on different scheduling policies may need to be generated because the choice of the appropriate scheduling policy for a given demand-resource scenario is not clear.

When scheduling algorithms which are not explicitly consistent with applicable scheduling policies are used, generation of multiple schedules with different scheduling heuristics, heuristic control parameters and/or algorithms is essential. Resulting multiple schedules must then be compared on the basis of the criteria relevant to the applicable scheduling policies. The following paragraphs describe those criteria, define metrics for comparing schedules based on those criteria, and describe the use of the metrics for comparing schedules based on sets of the criteria.

3. Evaluation of Alternate Schedules

Scheduling success rate (that is, percent or total number of requests scheduled) has been the traditional criterion for schedule effectiveness. Scheduling success rate is an adequate measure of schedule effectiveness when the initial scheduling policies are unimportant or do not affect the resulting schedule, for example, in a demand-resource scenario in which every request is successfully scheduled at the requested duration. However, scheduling success rate alone does not adequately measure schedule effectiveness under demand-resource scenarios that requires rejection or scheduling of many requests at reduced durations.

For example, consider two schedules: Schedule X and Schedule Y. Figure 1 shows plots of the cumulative number of requests scheduled versus priority for the two schedules. Schedule X is biased in favor of higher priority users, whereas Schedule Y is biased in favor of lower priority users. Clearly, Schedule Y should not be selected if the primary scheduling policy is to maximize number of scheduled events in priority order. If the policy is to maximize the scheduled requests, then Y, which has more requests scheduled than X, should be selected.

Which scheduling policy is appropriate for a given demand-resource scenario? Acceptability of a schedule often depends on multiple schedule effectiveness criteria, even though the initial scheduling policy implies a unique schedule effectiveness criterion. Some criteria relate directly to the various initial scheduling policies; the rest relate to other desirable qualities in an effective schedule.

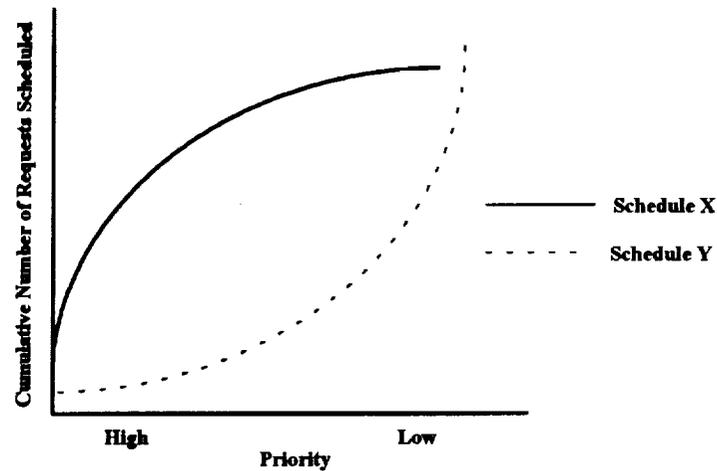


Figure 1. Schedule X Satisfies Priorities Better Than Schedule Y

3.1 Evaluation Criteria

The criteria for schedule evaluation discussed in this paper are categorized as follows:

- Maximizing success rates in priority order
- Satisfying specified success rates in priority order
- Balancing success rates among customers
- Minimizing the impact of undesirable gaps between successive scheduled requests

None of these criteria involves the success of any specified individual requests. Specific requirements for the success of individual requests are assumed to be ensured by means of appropriate algorithm and problem definition, that is, correct definition of requests and enforcement of constraints.

3.1.1 Maximizing Success Rates in Priority Order

Scheduling success may be defined as the percentage of events scheduled or as the percentage of requested time scheduled. Maximizing success rates in priority order means maximizing the success rate of a given customer without considering the success rates of lower priority customers.

These definitions lead to two distinct criteria:

- Maximization of event success rates in priority order
- Maximization of requested time success rates in priority order

For schedule comparisons based on one of these criteria, comparing the success rates in priority order is sufficient. A schedule that has the maximum success rate for the highest priority best satisfies the criterion. However, strict adherence to only one criterion could result in failure to consider schedules that may be nearly as good for the highest priority and superior for lower priorities. Table 1 shows three sample schedules, with the number of events requested and scheduled for each priority.

Priority	Number of Events Requested	Number of Events Scheduled		
		Schedule A	Schedule B	Schedule C
1	100	90	91	80
2	100	80	75	85
3	100	60	65	65
4	100	45	40	40

Table 1. Schedule Statistics by Priority for Three Sample Schedules

Comparing the schedules strictly on the basis of maximization of success rates in priority order results in selecting schedule B to be the best: Schedule B has 91 priority-1 scheduled events, as compared to 90 priority-1 scheduled events for Schedule A. However, such a comparison does not indicate the nearness of Schedules A and B, and it fails to recognize that Schedule A has gained five priority-2 events and lost only one priority-1 event and five priority-3 events. Such gains and losses between adjacent priorities among different schedules make the near equivalence among schedules difficult to recognize from the raw information (as shown in Table 1), especially when the number of priorities is large. Near equivalence of schedules is more easily recognizable from a table of cumulative events scheduled in priority order. Table 2 presents the cumulative events scheduled for the example shown in Table 1.

Priority	Cumulative Number of Events Requested	Cumulative Number of Events Scheduled		
		Schedule A	Schedule B	Schedule C
1	100	90	91	80
2	200	170	166	165
3	300	230	231	230
4	400	275	271	270

Table 2. Cumulative Events Scheduled, by Priority for Sample Schedules A, B, and C

Table 2 shows the following:

- Schedules A and B are nearly the same and Schedule C is substantially worse when the number of events scheduled for priority 1 alone are compared.
- Schedule A is substantially better than Schedules B and C when cumulative number of events scheduled for priorities 1 and 2 are compared.
- All three schedules are nearly the same when cumulative number of events scheduled for priorities 1 through 3 are compared.
- Schedule A is substantially better than Schedules B and C when cumulative number of events scheduled for priorities 1 through 4 are compared.

This analysis suggests that schedule A is either nearly the same as or better than Schedules B and C at every priority level and that Schedule A is better than Schedule B, even though Schedule B best satisfies maximization of event success rates in strict priority order. Figure 2 presents the information from Table 2 in graph form and illustrates the desirability of Schedule A over Schedules B and C.

The preceding analysis led to a definitive conclusion on a simple example with four priorities and three schedules. However, it would be difficult to come to any definitive conclusion using tables like Table 2 and charts like Figure 2 when the number of priorities and the number of schedules are large. The chart would have crisscrossing plots, rendering visual recognition of an overall better schedule a difficult task. Furthermore, this type of analysis based on subjective judgment is

not suitable for automated comparison of schedules. Automated comparison of schedules based on maximization of success rates in priority order requires a more comprehensive method involving numerically computable evaluation metrics. The following few paragraphs suggest such metrics.

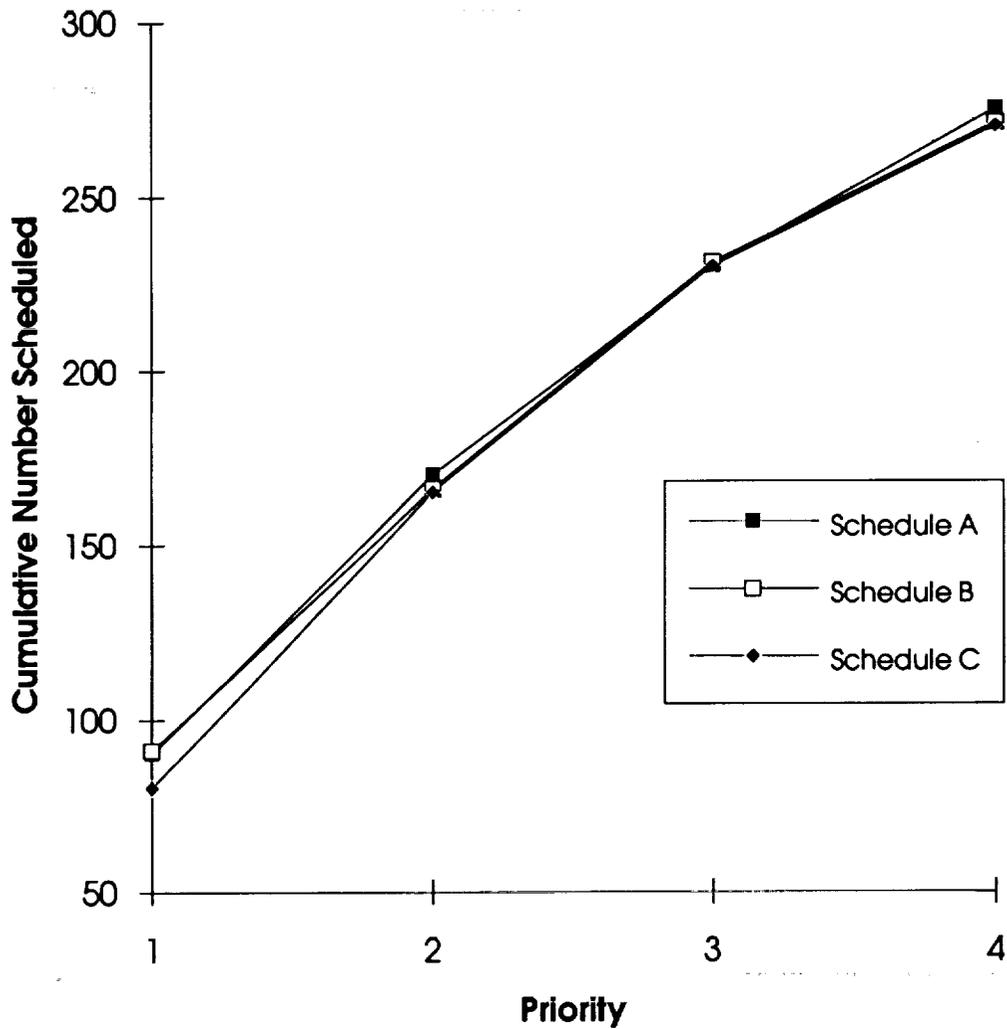


Figure 2. Cumulative Number of Events Scheduled Versus Priority

Maximization of event success rates in strict priority order implies that each event for a given priority is *more important* than all lower priority requests combined. In this scenario, request for a given priority has a value that is the sum of the values of all requests for lower priorities. If v_i is the value of an i th priority event, and n_i is the number of requested i th priority events, then

$$v_p = 1$$

$$v_i = \sum_{k=i+1}^p v_k \cdot n_k \text{ for } i < p$$

Values computed in this way grossly overstate the differences in values of events of adjacent priorities; they also disallow compensating for any loss of higher priority events with a gain in the number of lower priority events scheduled (as in the example.) In reality, the differences in relative values of events of adjacent priorities are much smaller than those given here.

The true differences can be expected to be consistent with compensation of loss for higher priority events by a larger gain for adjacent lower priority events. A practical comparison of schedules should be based on values of events that more reasonably represent the true differences between adjacent priorities, without understating the differences between widely separated priorities. Setting event values proportional to the total number of requested events for lower priorities is suggested here as a more reasonable representation of the true differences between values of events of different priorities. Mathematically, if N_i is the value of an i th priority event, and n_i is the number of requested i th priority events, then

$$N_p = 1$$

$$N_i = \sum_{k=i+1}^p n_k \text{ for } i < p$$

where

p = number of priorities

Therefore, the following metric, based on the weighted sum of scheduled events with weights equal to N_i , is a reasonable suggestion for comparing schedules based on maximization of scheduled event success rates in priority order:

$$\rho = \frac{\sum_{i=1}^p s_i N_i}{\sum_{i=1}^p n_i N_i}$$

where

ρ = degree of maximization of event success rates in priority order

s_i = number of requests scheduled for i th priority

The denominator in the formula represents the maximum possible value of a schedule and is used to limit the value of the metric ρ between 0 and 1.

Similarly, assuming that the value of a unit of requested time for a given priority is proportional to the combined requested time for all lower priorities, the metric for comparing schedules based on maximization of requested time success rates in priority order can be defined as:

$$\tau = \frac{\sum_{i=1}^p z_i T_i}{\sum_{i=1}^p t_i T_i}$$

where

τ = degree of maximization of requested time success rates while enforcing priorities

z_i = time scheduled for i th priority

t_i = time requested for i th priority

$T_p = 1$

$T_i = \sum_{k=i+1}^p t_k$ for $i < p$

Schedules A, B, and C from Table 1 by applying the previously defined metric ρ for maximization of event success rates in priority order produces the schedule rankings shown in Table 3. The resulting rankings clearly indicate that Schedules A and B are nearly equally preferable; however, the slightly higher metric value of Schedule A appears to have captured the advantage of the additional five priority-2 events over the loss of a single priority-1 event.

	Schedule A	Schedule B	Schedule C
Value of ρ	0.816	0.812	0.791
Rank	1	2	3

Table 3. Metric Values and Rankings of Sample Schedules A, B, and C

3.1.2 Satisfying Specified Success Rates in Priority Order

The metrics for comparing schedules using the criterion of satisfying specified success rates in priority order can be based on a rationale very similar to that used for the metrics for maximization of success rates in priority order.

In terms of satisfying specified event success rates in priority order, scheduling more than the specified event success rate for any priority has no value. That is, any events scheduled over and above the minimum specified success rates have *zero* values. Therefore, the maximum number of events that must be scheduled to meet the criteria for i th priority with a specified success rate of q_i is $n_i q_i$. Assuming that a value W_i of an i th priority event is proportional to the number of events required to satisfy the specified success rates at all the lower priorities,

$$W_p = q_p$$

$$W_i = \sum_{k=i+1}^p n_k q_k \text{ for } i < p$$

where

$$0 \leq q_i \leq 1$$

The following metric based, on the weighted sum of the *nonzero*-value scheduled events with weights equal to W_p , is a reasonable suggestion for comparing schedules based on satisfying specified event success rates in priority order:

$$\gamma = \frac{\sum_{i=1}^p \text{Min}(s_i, n_i q_i) W_i}{\sum_{i=1}^p n_i q_i W_i}$$

where

γ = degree of satisfying specified event success rates in priority order

Similarly, from the perspective of satisfying specified time scheduled success rates in priority order, scheduling more than the specified requested time success rate for any priority has no value. That is, any time scheduled over and above the minimum specified success rates has *zero* value. Therefore, the maximum time that must be scheduled to meet the criteria for i th priority with a specified success rate of q_i is $t_i q_i$. Assuming that a value U_i of an i th priority event is proportional to the time required to satisfy the specified success rates at all the lower priorities,

$$U_p = q_p$$

$$U_i = \sum_{k=i+1}^p t_k q_k \text{ for } i < p$$

The metric, based on the weighted sum of the valuable (*nonzero* value) scheduled events with weights equal to U_i for comparing schedules based on satisfying specified scheduled time success rates in priority order is

$$\varphi = \frac{\sum_{i=1}^p \text{Min}(z_i, t_i q_i) U_i}{\sum_{i=1}^p t_i q_i U_i}$$

where

φ = degree of satisfying specified scheduled time success rates in priority order

3.1.3 Balancing Success Rates Among Customers

The scheduling criterion of balancing success rates among customers implies a goal to generate a schedule with equal success rates for all customers. This criterion, when applied literally, is fully satisfied even by a schedule with zero success rates for all customers. Hence, this criterion should be interpreted to imply balanced success rates among customers with maximum combined success rate of all customers.

The combined event success rate c of a schedule can be calculated as

$$c = \frac{\sum_{i=1}^p s_i}{\sum_{i=1}^p n_i}$$

The success rates for all customers are balanced when the success rate for each customer equals c . Therefore, scheduling more events for a customer than are necessary to provide a success rate of c has *zero* value from the standpoint of balancing event success rates. Furthermore, scheduling additional events for a customer with a success rate greater than or equal to c could have an adverse impact on customers with success rates lower than c . Therefore, an event scheduled for any customer over and above the required number to provide a success rate of c can be assumed to have *zero* value from the standpoint of balancing event success rates. Hence, the number of *nonzero-value* events scheduled for the i th priority customer can be calculated as $Min(s_i, cn_i)$. The total number of *nonzero-value* events scheduled can be calculated as

$$\sum_{i=1}^p Min(s_i, cn_i)$$

It naturally follows that the total number of *nonzero-value* events scheduled is a reasonable basis for a metric to compare the schedules based on the criterion of balancing event success rates among customers. Hence, the metric α for comparing schedules based on the criterion of balancing event success rates among customers can be defined as

$$\alpha = \frac{\sum_{i=1}^p Min(s_i, cn_i)}{\sum_{i=1}^p n_i}$$

From the perspective of balancing time scheduled success rates, scheduling more time for a customer than necessary to provide a success rate d , where

$$d = \frac{\sum_{i=1}^p z_i}{\sum_{i=1}^p t_i},$$

has a *zero* value from the standpoint of balancing time scheduled success rates. Therefore, the total *nonzero* value time scheduled can be calculated as

$$\sum_{i=1}^P \text{Min}(z_i, dt_i)$$

As a result, the metric β for comparing schedules based on the criterion of balancing time scheduled success rates among customers can be defined as

$$\beta = \frac{\sum_{i=1}^P \text{Min}(z_i, dt_i)}{\sum_{i=1}^P t_i}$$

3.1.4 Minimizing the Impact of Undesirable Gaps Between Successive Scheduled Requests

A schedule that includes an event for every customer request is obviously the schedule that best satisfies the customer. The time gaps between each consecutive pair of events (i.e., between events 1 and 2, 2 and 3, 3 and 4, and so forth) in such a schedule are within the maximum gap size implicitly intended by the customer, based on the requested start times and tolerances. When some of the customer's requests are not scheduled, these gaps exceed the customer's intended maximum gap size. Figure 3 shows an excessively long gap could mean loss of data for the customer.

A customer is less likely to have adverse impacts when the customer's scheduled events and declined events are more evenly interspersed in time than when otherwise. For example, a customer who has requested six events is more likely to have an adverse impact when the last three requested events are declined than when every other requested event is declined, even though the number of declined events is three in both cases. Generally, most customers can endure one declined event between two scheduled events without a significant loss of data; however, customers who have two or more declined events in succession are likely to experience a significant loss of data. Therefore, the amount of data loss in an excessively long gap can be expected to be proportional to the length of the gap in excess of twice the sum of the duration and the intended gap. Further, the duration of the event may be assumed to represent the amount of data transmitted during the requested event.

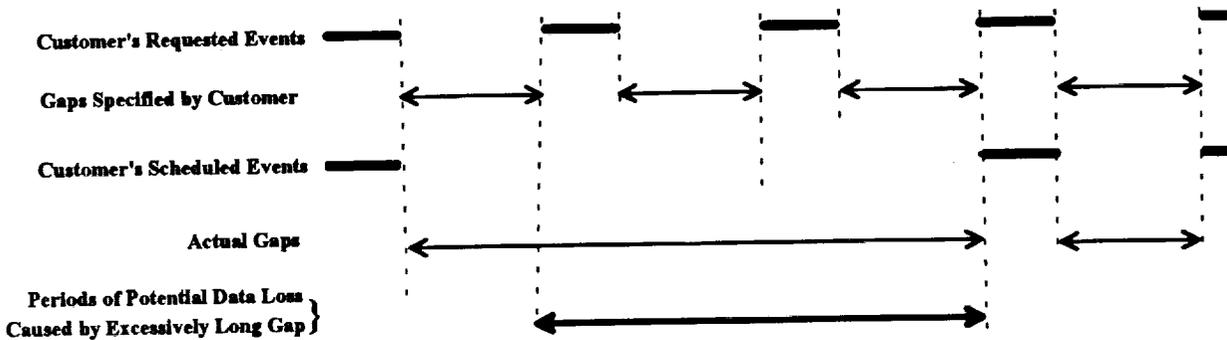


Figure 3. Potential Loss of Data Because of Excessively Long Gaps

Mathematically, if t is the average duration of the requested events for a customer, g the maximum gap intended by the customer, and e the length of the gap between any two successive scheduled events for the customer, the loss of data l because of gap e can be approximated by

$$l(e) = \frac{\text{Max}\{0, e-2(g+t)\}t}{(g+t)}$$

This formula can be used to estimate the data loss associated with each gap for each customer. The total data loss associated with a schedule, λ , is estimated by summing the data losses associated with all gaps for every customer. Among all schedules, the schedule having the minimum total data loss is the most desirable.

3.2 Schedule Evaluation Example

Metrics defined in the preceding sections were used to evaluate schedules for a scenario representative of the expected 1998 SN demand, assuming availability of 4 single access antennas. Table 4 summarizes the demand scenario which represents support of one week's demand for 11 customers requesting a total of 1,555 events and 42,084 minutes. All the unmanned flight customers are assumed to have duration flexibility. In addition, some customers are assumed to have start time flexibility. The flexibility parameters used in the scenario are based on discussions with the associated customers and, therefore, are representative of the true flexibility available to these customers.

Customer	Priority	Events Requested	Nominal Dur. (Min)	Minimum Dur. (Min)	Flexibility (%)	Requested Time (Min)	Specified Succ. Rate(%)
STS-121	1	208	50	50	0	9,119	90
SSFreedom	2	202	50	50	0	9,305	60
SIRTF	3	195	22	12	0	4,290	60
AXAF-S	4	194	22	12	0	4,268	60
EOS-AM1	5	218	15	10	67	3,270	60
HST	6	195	18	15	0	3,510	60
GRO	7	52	30	16	40	1,560	60
TOPEX	8	28	18	14	100	504	60
XTE	9	50	25	15	100	1,250	60
Landsat-7	10	109	24	14	0	2,616	60
TRMM	11	104	23	20	87	2,392	60
Total		1,555				42,084	

Table 4. Demand Scenario for Schedule Evaluation Example

Seven different schedules, P through V, were generated for the example using different scheduling strategies. Computer Sciences Corporation's Space Network Scheduling Prototype generated the schedules. Table 5 shows the number of events scheduled for the seven schedules. Table 6 shows the amount of time scheduled for the seven schedules.

Table 7 shows the values of metrics for the seven different schedules for the example. Table 8 shows the ranking of the seven schedules for each of the metrics. As is apparent from Table 8, different schedules are ranked number 1 for different metrics. For example, Schedule U is the best schedule in terms of maximizing events in priority order (ρ), whereas Schedule Q is the best in terms of maximizing scheduled time in priority order (τ). Even though Schedule Q maximizes scheduled time in priority order, it ranks very low in terms of maximizing the number of events scheduled (see Table 8).

Customer	Schedule						
	P	Q	R	S	T	U	V
STS-121	206	207	192	189	143	207	207
SSFreedom	202	202	186	185	140	202	202
SIRTF	109	152	141	148	146	187	164
AXAF-S	106	154	151	150	157	182	166
EOS-AM1	119	140	148	155	175	214	188
HST	66	92	117	104	149	155	133
GRO	7	27	34	28	40	47	35
TOPEX	25	25	25	26	26	28	28
XTE	25	36	42	39	47	47	44
Landsat-7	14	45	58	56	89	89	69
TRMM	15	47	63	59	87	91	69
All	894	1127	1157	1139	1199	1449	1308

Table 5. Number of Events Scheduled for the Example

Customer	Schedule						
	P	Q	R	S	T	U	V
STS-121	9057	9090	8372	8282	5991	7334	8617
SSFreedom	9305	9305	8506	8476	5972	7237	8838
SIRTF	2398	3111	2844	3058	2998	2930	2744
AXAF-S	2332	3119	2993	3081	3200	3104	2835
EOS-AM1	1785	2010	2138	2245	2513	2696	2394
HST	1188	1524	1977	1746	2537	2318	1935
GRO	210	608	817	680	981	932	764
TOPEX	450	446	443	464	460	460	462
XTE	625	815	951	906	1086	922	850
Landsat-7	336	820	1076	1057	1882	1567	1300
TRMM	345	808	1147	1100	1754	1626	1203
All	28031	31656	31264	31095	29347	31126	31942

Table 6. Time Scheduled for the Example

Metric		Schedule						
		P	Q	R	S	T	U	V
Maximize events in priority order	ρ	0.7411	0.8438	0.8124	0.8102	0.7400	0.9645	0.9072
Maximize time in priority order	τ	0.8388	0.8874	0.8327	0.8328	0.6761	0.7677	0.8512
Satisfy specified event success rates in priority order	γ	0.9509	0.9906	0.9999	0.9949	0.8518	1.0000	1.0000
Satisfy specified scheduled time success rates in priority order	ϕ	0.8470	0.8932	0.8372	0.8368	0.6751	0.7697	0.8561
Balance event success rates	α	0.4584	0.6343	0.6871	0.6716	0.7460	0.9018	0.7910
Balance scheduled time success rates	β	0.5186	0.6428	0.6649	0.6620	0.6741	0.7085	0.6746
Loss of data	λ	4210	1119	3693	4327	14775	230	397

Table 7. Evaluation Metrics for Schedule Evaluation Example

Metric		Schedule						
		P	Q	R	S	T	U	V
Maximize events in priority order	ρ	6	3	4	5	7	1	2
Maximize time in priority order	τ	3	1	5	4	7	6	2
Satisfy specified event success rates in priority order	γ	6	5	3	4	7	1	1
Satisfy specified scheduled time success rates in priority order	ϕ	5	4	1	3	7	6	2
Balance event success rates	α	7	6	4	5	3	1	2
Balance scheduled time success rates	β	7	6	4	5	3	1	2
Loss of data	λ	5	4	2	6	7	1	3

Table 8. Schedule Rankings Based on the Evaluation Metrics for the Example

Therefore, selecting the best schedule depends on whether a single criterion or multiple criteria are to be used. Multiple criteria requires a more complex selection procedure. First, weights proportional to the level of importance must be assigned to each relevant criteria. Then, the schedule that minimizes the weighted sum of ranks is the most satisfactory schedule, based on the selected multiple criteria.

For example, if it is equally important to maximize time (τ) and maximize events (ρ) in priority order, then the weights assigned for maximizing events in priority order and for maximizing scheduled time in priority order should each be set at 0.5. Schedules Q and V are the best schedules in this case. Because in this case Schedules Q and V are equivalent from the perspective of maximizing events in priority order and maximizing scheduled time in priority order, the two schedules could be further examined on secondary criteria for example, balanced success rates. In such a case, Schedule V would be superior to Schedule Q.

4. Conclusion

This paper has described a formal mathematical procedure which allows automated comparison of schedules based on scheduling policy specific criteria.

Evaluation of schedules based solely on the total number of scheduled events has been shown to be inadequate to fulfill the current SN priority order scheduling policy. As the demand for SN resources increases, evaluation of schedules based on total number of scheduled events will become even more inadequate and the SN will need to use alternative criteria. This paper identified several alternative scheduling policies and derived schedule evaluation metrics which are directly related to those policies as well as the currently used priority order scheduling policy. It described a method to use these metrics in comparing schedules based on multiple criteria. The method which is suitable for automated schedule comparison involves the following steps:

- Calculation of the metrics for all possible criteria (α , β , γ , etc.)
- Ranking the schedules based on each of these criteria
- Assigning weights to each of these criteria based on relative importance each criterion
- Finding the weighted sum of the criteria specific ranks
- Selection of the schedule which minimizes the weighted sum of ranks.

N 9 4 - 2 1 3 3 7

**THE TEST ANALYSIS RETRIEVAL SYSTEM (TARS)
MEETING THE CHALLENGES OF
THE NETWORK'S TEST PROCESSES**

Prepared By:

**Robert L. Stelmaszek and Douglas R. Lumsden
NASA/GSFC 531.4 Loral/GSFC 531.4**

1.0 Summary

The Networks Systems Test Section (GSFC 531.4) is responsible for managing a variety of engineering and operational tests used to assess the status of the Network elements relative to readiness certification for new and ongoing mission support and for performance trending. To conduct analysis of data collected during these tests, to disseminate and share the information, and to catalog and create reports based on the analysis is currently a cumbersome and inefficient task due primarily to the manual handling of paper products and the inability to easily exchange information between the various Networks elements.

The Test Analysis and Retrieval System (TARS) is being implemented to promote concise data analysis, intelligible reporting of test results, to minimize test duplication by fostering a broad sharing of test data, and perhaps most importantly, to provide significantly improved response to the Network's internal and external customers. This paper outlines the intended application, architecture and benefits of the TARS.

2.0 Introduction

The Networks Test Section of the Telecommunication Systems Branch in the Goddard Space Flight Center's Networks Division manages the development and conduct of a variety of tests and simulation activities for the purpose of ensuring continuing compliance with all Networks internal and external, space and ground based interfaces. This responsibility also includes all Networks readiness activities for support of upcoming missions as well as routine characterization of systems for performance trending evaluations.

- The Challenge

The participants for the conduct of these test and simulation activities can vary from strictly on-site personnel such as for Station Readiness Tests, to a globally diverse team typically involved in network end-to-end tests. In all cases the Network's control point is the GSFC Network Control Center (NCC) staffed by a Test Director (TD) who manages the end-to-end coordination and test conduct. One of the TD's responsibilities is to generate a written test report which is supplemented by reports from each participating test element. Ideally, these various inputs are synthesized into a comprehensive set of data and conclusions in harmony with each participant's assessment. This, however is not always the case due to the cumbersome process of creating, collecting and summarizing various paper reports containing test results information.

During the planning phase for these tests there is rarely any research done to assess if data collected during previously run tests could apply to the test being planned to potentially negate the need for even running the test. This is largely because of the difficulties inherent in searching paper files for data which may or may not exist. Clearly significant amounts of valuable Networks resource time could be saved with the ability to utilize sets of test data to satisfy multiple needs.

- The Solution

The Test Analysis and Retrieval System (TARS) is being implemented to meet the challenges of the Network test processes. The purpose of the Test Analysis and Retrieval System (TARS) is to convert today's paper intensive data collection, archival and dissemination environment into a work station based electronic media system which will facilitate ease of access to test data for on-line analysis, trending, and report generation. Once the first phase of TARS is established locally at GSFC it is planned to be augmented to extend its availability to other Networks facilities involved with testing such as the White Sands Complex. A number of existing informational data bases will then be integrated with TARS to provide a comprehensive archive of data from, for example, TDRS satellite performance tests which are used for end-of-life trending forecasts.

3.0 The Test Analysis Retrieval System (TARS)

3.1 TARS Overview

TARS is a database driven system which stores and enables access to test data and results of various network testing. The system provides common access to a shared database and other software on a Netware fileserver. The objective of TARS is to convert the current paper intensive test data environment to an electronic system utilizing state-of-the-art 10Base-T Ethernet LAN technology and a relational database configuration which facilitates on-line sorting and searching of test data using keywords and relationships.

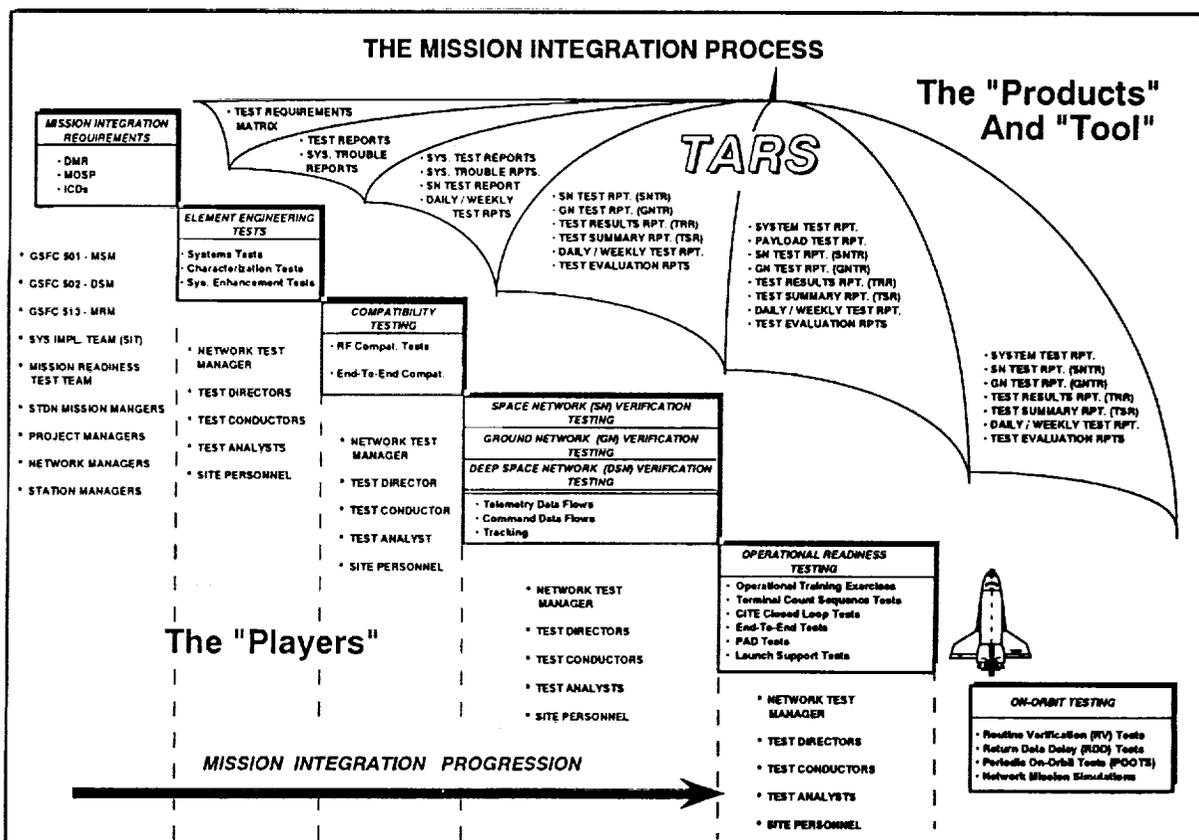
TARS is being developed by the Mission Operations and Data Systems Directorate's (MO&DSD's) Systems Engineering and Analysis Support (SEAS) contractors specifically for the Networks Test Section Code 531.4. The system is designed to comprise a collection of available Commercial-Off-The-Shelf (COTS) hardware and software with a minimum of specially designed software integrated into a system able to assemble and control the results of network test data. TARS will create test reports, provide system users with easy access to archived data, and perform analysis and comparisons of specific data types useful in assessing performance curves. A primary goal of the system implementors is to assure efficiency, effectiveness, and user friendliness for it's various users.

TARS consists of three major hardware components, the file server, work stations, and the LAN. TARS is a relational database driven system which will initially perform the five major functions of data collection, data storage, data manipulation, data distribution and data display. Additional TARS capabilities including assembly of characterization test data and plotting and graphing of engineering information in hard copy form are planned.

3.1.1 TARS Mission Integration Test Support

The TARS will support many facets of the GSFC MO&DSD network mission integration and testing efforts. Network mission integration is a progressive process of certifying that the network resources are ready and qualified to support a particular mission. Figure 3.1.1 depicts the mission integration chronology including the tests, activities, and users which TARS is designed to support.

FIGURE 3.1.1 TARS AND THE MISSION INTEGRATION PROCESS



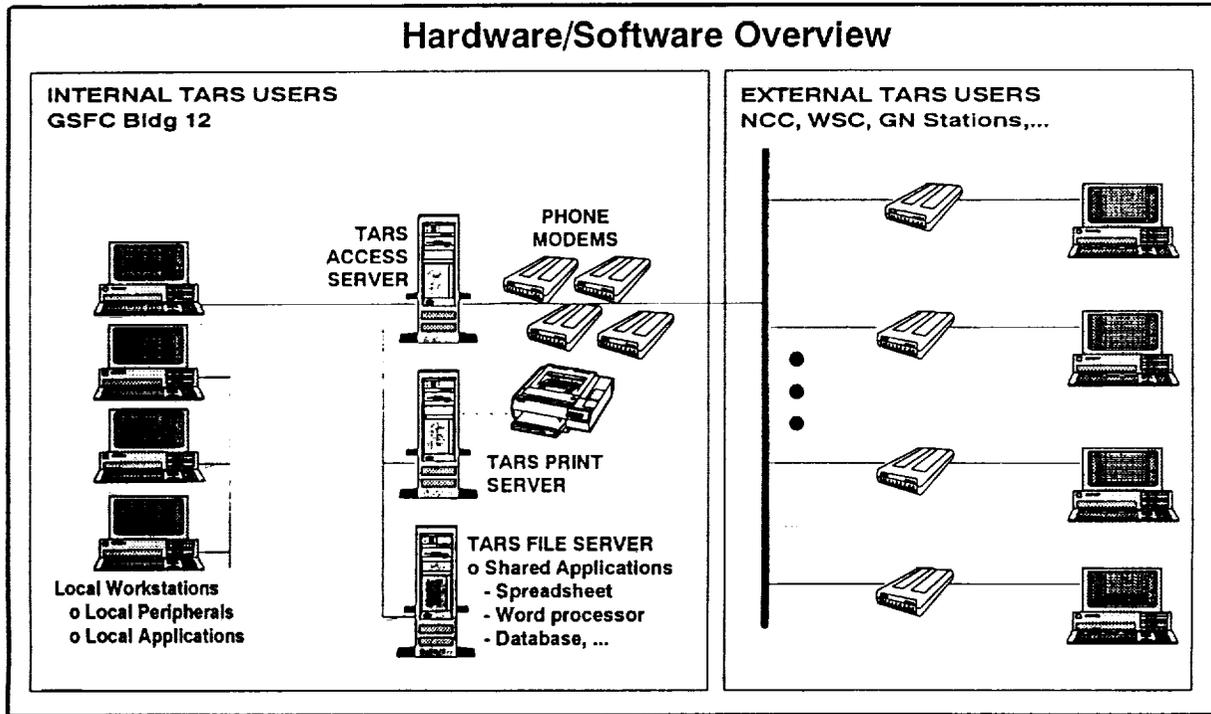
Network mission integration begins when network support and test requirements are levied on the Network by the projects, or mission managers. These requirements are managed and controlled by the MO&DSD Mission Support Managers (MSMs), Mission Readiness Managers (MRMs) and the Data Systems Managers (DSMs). These requirements will be entered into the TARS database and matrixed to the corresponding test activities to assure test requirements traceability and consummation.

The primary function of TARS is in support of the mission integration and test processes. This process typically begins with element engineering testing, progressing to compatibility testing, ground and space network verification testing, operational readiness testing and training, and culminating with on-orbit testing. Throughout this mission integration "life cycle" the TARS allows common access to a shared data base and software on a network file server. Through an existing local area network, TARS is designed to archive and assemble the results of the test data, generate plotting information, and create graphics and text of the test data in a hard copy or screen format. TARS will enable historical test data retrieval (search and sort), test analysis, statistical comparisons, and data formatting for test technicians/ engineers, management and the space network users.

3.1.2 The TARS Users

TARS has potentially many users. These users can be classified as internal and external users. For the purpose of this paper the internal users are defined as on-site GSFC Building 12 test personnel who have access to TARS via the local area network and have direct on-line access to the TARS access server. The external users are those test personnel that are located off-site or outside of GSFC Building 12. Such external users include the NCC, and the SEAS and NMOS contractor facilities. The external users are planned to include off-site Network locations such as the White Sands Complex (WSC) and various Ground Network (GN) stations and potentially the Deep Space Network (DSN) stations. Details of the TARS user capabilities and potentials are described in the succeeding paragraphs. Figure 3.1.2 depicts the general TARS user support configuration.

FIGURE 3.1.2 TARS USER SUPPORT CONFIGURATION



3.1.2.1 Internal User

The primary internal user of the TARS is the GSFC Mission Operations and Data Systems Directorate's Network Test Section. The TARS allows the Network Test Section personnel to archive test data and retrieve the data for analysis and reporting. TARS enables the Network Test Section personnel to review engineering test requests and compare them with archived data to assure that tests have not been previously performed and/or that the data is not currently available. If the testing involves a unique test scenario the system will provide data which assures that test plans and briefing messages will accomplish the desired results and can be coordinated with the WSC and other test participants.

- Network Test Manager (NTM)

The Network Test Manager (NTM), is the mission verification lead for Network Test Section. The NTM utilizes TARS to store and to trace network test requirements and to assure that those requirements are met either through analysis or testing. As tests are performed the NTM assures that all pertinent test data is gathered and archived in the TARS. The test data includes test characteristics, test article and spacecraft parameters, discriminators, and test issues and discrepancies. The NTM will use data in TARS to assess and analyze the test results.

- Test Conductors and Test Analysts

The network test conductors and test analysts will use the TARS to store test data for future or real-time analysis. The test conductors will use TARS to develop comprehensive test results reports and analysis studies. The system will be used to transform test data into tabular, graphic, and narrative reports. Test analysis will be conducted using report output or through "on-screen" data display and manipulation.

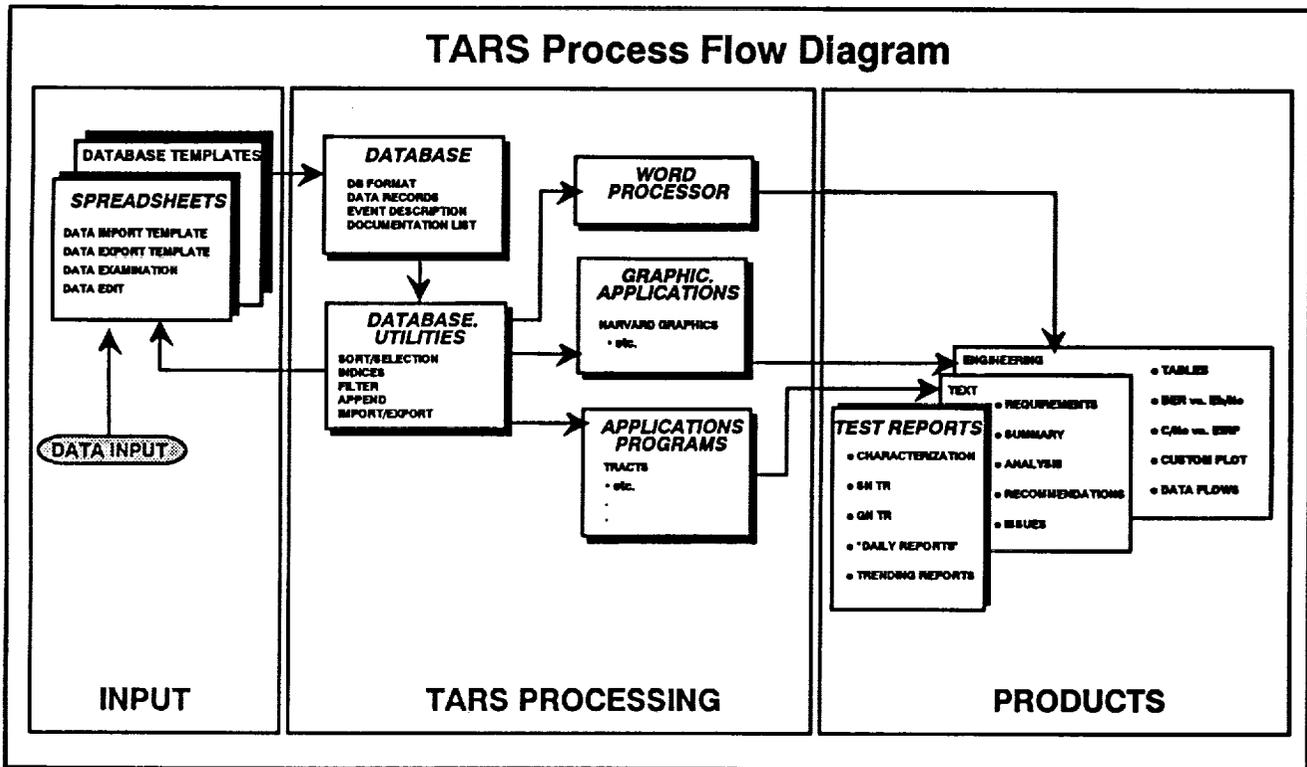
3.1.2.2 External User

The external users will use TARS primarily for test data retrieval, evaluation and analysis, and as planned, will use it for data input and test reporting. Network site personnel (i.e. NCC, WSC, GN, DSN sites, etc.) will eventually have the ability to access the TARS database directly via a site terminal and a modem. This system will enable test data input/archiving and will provide the site personnel with real-time access to test data, test analyses and reports, and it will enable accurate research and evaluation of previous test runs.

3.1.3 TARS Process Description

The following sections will describe the general structure of TARS describing it's components in terms of functionality and providing a general description of the specific products and applications. Figure 3.1.3 illustrates the TARS process flow.

FIGURE 3.1.3 THE TARS PROCESS FLOW



TARS 09

3.1.3.1 Data Input, Storage and Manipulation

Test Data is input to TARS through the use of either a spread sheet or directly via a database template. The compatible spreadsheets include Lotus 1-2-3, Quattro or Excell. These spreadsheet applications provide the framework for the entry and sorting of test data. Data import and export templates will be used to transfer the data from the spreadsheet to the database format. A database template will also be available directly to enter the data into the relational database. The TARS FoxPro database aids the user in insuring the accuracy of the data by checking the data characteristics (i.e. number of words, field type, decimal placement, maximum and minimum values, etc).

The system stores information in the form of data files. Although it is possible to organize the contents of data files in a wide variety of ways, it is important to note that TARS uses a relational database structure which resides on a LAN file server rather than a "flat" database. This relational database not only enables the user to store data but also to change, find (search, select), rearrange (sort, prioritize, categorize), analyze, and relate/compare data in the data base.

The relational database files within TARS will contain infinite sets of records. Each record will be further divided into fields that can hold various types of data. This type of data organization provides a table of information where the horizontal rows of the table represent the records and the vertical columns represent the fields. For example, the TARS application uses a relational database file to store the characteristic data of a particular test run. Each record in the file will contain various types of information for the particular test run and equipment/systems, while each field in the file contains the same type of information for all test runs. Figure 3.1.3.1 provides a typical example of a portion of a space network test data file.

Similar to the columns of a table or index, the fields can be of various lengths, and can hold various types of data. A TARS test data file typically contains fields representing dates, numeric values (ie. data rates, decibels, frequencies, channels, ratios), logical values (ie. True/False, Yes/No), and fields representing strings of characters such as test descriptions, resources titles, and analysis/evaluation statements. Like the rows of a table, each record has the same set of fields (although the contents of the field vary). The contents of the fields on a particular record, like the contents of a particular row on a table, are related to one another and this relationship is the origin of the term relational database.

Core programs invoke functions such as data analysis, and report writing. These programs read their data directly from files that the host FoxPro interpreter understands, and generates their results in the form of FoxPro files. As a result, most types of data manipulation - data input, modification, queries, simple reports - can be accomplished in FoxPro, which provides a particularly rich and open-ended set of commands with which to manipulate data. The use of the FoxPro application makes TARS a highly flexible and user-definable test analysis and test data retrieval system.

The TARS performs queries or searches on multiple parameters. These parameters include, but are not limited to, test dates, test data (for characterization tests), results of EIF testing (Go/No Go), test personnel/conductors, test location, test type, test levels, test issues/problems, and various data results.

3.1.3.3 Graphics Capabilities and Report Products

TARS will use several methods of graphics and report generation. The primary application to generate graphs of test data is Harvard Graphics. Typically a user will access TARS to "interrogate" the archived data; retrieve the pertinent historical test data based on the desired query, sort, and/or select criteria; and then import the data or files to his/her local system. Once the data is resident on the user's local system, he has the ability to generate graphics, charts, plots, etc. using the available graphic applications on his local system. In addition, reports and analysis text will be generated using standard word processor applications in conjunction with database ASCII files and tabular printouts.

The products which the TARS generates or facilitates will be associated with the life cycle of the integration and test process for a particular mission or system implementation. Typically, the system will generate data files, graphs, and text needed to create reports. The products may range from system trouble reports to weekly test reports to network test and result reports. The TARS will also be used to support various system analyses, studies and evaluations. Figure 3.1.1 above further illustrates some of the TARS products associated with the integration processes.

3.4 TARS Hardware Configuration

The TARS hardware configuration is a compilation of existing GSFC Code 531.4 resources and procured COTS hardware. The configuration consists of a local area network (LAN), work stations, a file server and access server, and modems.

The existing GSFC Code 531.4 10Base-T Ethernet LAN is in place and in use by test section NTMs, NCC test console and test analysis personnel. This LAN will be augmented to interconnect the TARS work stations and the TARS fileserver using Novell Netware software.

The TARS work stations will be configured with Microsoft DOS 5.0 or above, and Microsoft Windows 3.1. Figure 3.4. illustrates the minimum requirements for a typical TARS user terminal or work station.

FIGURE 3.4 THE TARS USER TERMINAL REQUIREMENTS

TARS WORKSTATION MINIMUM REQUIREMENTS	
<i>DOS PC</i>	<i>Macintosh</i>
386 SX	MAC PLUS
4 MB RAM	4 MB RAM
40 MB HARD DISK	40 MB HARD DISK
VGA MONITOR	VGA MONITOR
MOUSE	SYSTEM 7
ETHERNET CARD (INTERNAL USER)	ETHERNET CARD (INTERNAL USER)
MODEM (EXTERNAL USER)	MODEM (EXTERNAL USER)
WINDOWS 3.1	
DOS 5.0	

The file server is comprised of a full-height tower 486DX2/66 CPU with 286 KB - 25 nanoseconds of cache memory, and 16 MB - 60 nanoseconds of user RAM expandable to 32 MB. The server processor speed will be 500 to 550 MB SCSI 11 milliseconds. The server will also house a 1.0/2.0 GB DAT compatible tape drive with a software driver and backup application software for system routine and emergency backup capability.

4.0 Conclusion

The Test Analysis and Retrieval System (TARS) represents a major process improvement using the latest database technology, off-the-shelf equipments, and available efficiencies. This system will improve the access, archival, analysis, and reporting of the extensive and valuable data associated with the many Networks test and verification activities and processes. This system is a practical tool used to enhance the Networks data analysis and retrieval efficiency, effectiveness and productivity.



ANALYSIS OF THE TDRS MULTIPLE ACCESS SYSTEM
FOR POSSIBLE USE AS AN ATTITUDE CONTROL SYSTEM SENSOR

Bruce Allyn Blevins, Ph.D.
GTE Government Systems Corp.
White Sands Ground Terminal
Post Office Box 235
Las Cruces, New Mexico 88005

Victor J. Sank, Ph.D.
Fredrick Herold & Associates
7303A Hanover Parkway
Greenbelt, Maryland 20770

(505) 525-6911 (v)
(505) 525-6541 (f)

(301) 982-4001 (v)
(301) 982-4021 (f)

Summary

A member of the constellation of TDR satellites (TDRS) has experienced a failure of its prime earth sensor. Failure of the remaining earth sensor could result in the inability of the satellite to control its attitude and provide user services. Loss of the satellite would be a serious event. The multiple access (MA) antenna array on the TDRS has been proposed for use as a backup sensor for the attitude control system. This paper describes our analysis of the performance of the MA array as an interferometer used for accurate attitude determination.

A least squares fit of a plane to the MA phase information appears to represent the TDRS body roll and pitch within about 0.1° . This is sufficient for SGL pointing and MA and SSA user services. Analytic improvements that include ionospheric correction may yield sufficient accuracy for KSA user services.

Spacecraft Configuration

The Tracking Data and Relay Satellite (TDRS) is three axis stabilized and in geostationary orbit. The roll axis (X) points in the direction of orbital velocity, the pitch axis (Y) is perpendicular to the orbital plane with its sense in the south direction, and the yaw axis (Z) is toward earth nadir. Each TDRS contains two earth sensors (infrared sensors) and coarse and fine sun sensors (solar cells). When the satellite is in normal operation the earth sensor in use (the second one is a backup) is

scanned across the earth at a rate of 4 Hz along a line that is 5° above the equator. The earth disk as seen by the TDRS is $\pm 8.6^\circ$. The length of the scan line is an indication of roll and the position of the scan line relative to scan center is an indication of pitch. The earth sensor resolution is 0.01° and knowledge of the spacecraft roll and pitch attitude is maintained to 0.08° . Mechanically the roll and pitch attitude is controlled by changing the speed of two momentum wheels whose axes are in the YZ plane and whose net vector is in the negative Y direction.

Yaw attitude is maintained separately via the course and fine sun sensors and thrusters and is not considered in the paper.

For normal operations the space-ground-link (SGL) antenna must be pointed at the White Sands Ground Terminal (WSGT). Its antenna beamwidth is $\pm 0.375^\circ$. The MA forward beamwidth is $\pm 3^\circ$ and the MA return is $\pm 1.5^\circ$; thus, if the TDRS body can be maintained for SGL pointing, MA services can also be provided. The S-band Single Access (SSA) antenna beamwidth is $\pm 0.76^\circ$ and the K-band Single Access (KSA) beamwidth is $\pm 0.12^\circ$. In order to provide KSA services the TDRS body must be maintained to, or at least known to, $\pm 0.08^\circ$. If accurate roll and pitch attitude were not maintained, the TDRS would not be able to function.

In the mid 1970's the ATS-6 spacecraft was used to experiment with the concept of using antenna element phases for attitude control. Using the interferometer concept, the phase between two elements was measured onboard the spacecraft and sent to the ground via telemetry. Comparison of actual attitude, also sent in the telemetry, lead to the conclusion that RF phase differences received in separate antenna elements could be used for attitude control. Previous authors have proposed using the TDRS MA antenna system for attitude control (Reference 1). This is, however, much more complicated than the system used in the ATS-6 experiment.

The TDRS MA system is composed of 30 separate antenna elements with separate amplifiers, separate IF frequencies, and

30 separate SGL carrier frequencies. The 30 separately received signals are sent to the ground at 7.5 MHz increments over a 225 MHz band at Ku-band. On the ground the 30 different SGL carrier frequencies are converted to a common IF frequency (160 MHz at the WSGT and 29.75 MHz at the Second TDRS Ground Terminal (STGT)) so that they can be phase shifted and combined into a single signal for receiving a user spacecraft's data. For normal MA communications these phases are combined on the ground after several up and down conversions using several different local oscillators (LOs) both in the spacecraft and in the ground equipment. Even with all of the LOs referenced to the ground station common time and frequency standard (CTFS), it is not uncommon for phases to drift tens of degrees over a several hour period. An MA calibration emitter is placed at a known position on the earth (at the WSGT and the STGT) so that the system can be recalibrated (every 18 minutes for WSGT). For the MA communications service where the signals are phase shifted and summed, a 40° phase error for example, in half (15) of the elements would cause only a 0.5 dB degradation, which would not result in a serious service impact. On the other hand, a 40° phase error will cause a 0.4° error in attitude determination, which is unacceptable. The fact that the MA system works for communication services does not imply that it will be good enough for attitude control.

Attitude Control Modes

The usual attitude control modes for a standard TDRS after insertion in a geosynchronous orbit are: (1) Earth Mode, (2) Normal Mode, and (3) Sun Mode (Reference 2). The normal sequence for a TDRS after separation from the inertial upper stage is for the satellite to transition from earth mode to the normal mode. User services are provided when the satellite is in the normal mode. Upsets and stationkeeping can cause the satellite to transition from normal mode to earth mode and from earth mode to

sun mode. Detailed descriptions of the modes are presented below:

Normal Mode

During normal mode operation the pitch and roll attitude is determined by one of the earth sensors. Yaw attitude is determined by ground software; there is no onboard yaw control. Yaw attitude is constrained (not controlled) by the combination of angular momentum and yaw momentum drive interactions.

Usual operation of a TDRS is in the normal mode, in which the attitude control system is coupled directly to the earth sensors. Attitude is controlled by the reaction wheels which periodically require thruster firing for unloading when the momentum becomes excessive. Even though the earth sensor is scanning the earth at a 4 Hz rate, the attitude control system requires earth roll and pitch updates at one per second and averages every two updates.

Sun Mode

Sun mode operation has pitch and yaw attitudes determined by the coarse sun sensor. Rates for all axes are determined by three of the four gyros which are turned off due to lifetime constraints when in normal mode. With the solar arrays positioned at 90° or 270°, the plus or minus X-axis is pointed at the sun. The reaction control wheels (momentum wheels) are allowed to run down, and a roll rate of 0.12°/sec to 0.25°/sec is imposed around the X-axis to provide an earth sensor sweep search mode to locate the earth. Position and rates are controlled by the reaction control system (thrusters). This mode is used for safe storage and in preparation for recovery from loss of Earth reference.

Typical transition from the sun mode to the earth mode occurs at about 6:00 a.m. or p.m. (local spacecraft time). When the earth sensor detects the earth, ground commands stop the roll rotation about the X-axis, and attitude control is established to

keep the body of the spacecraft such that the earth nadir is normal to the MA array on the satellite. The earth sensor continually monitors the earth, provides roll and pitch information in the S-band telemetry for attitude control.

Earth Mode

In earth mode the pitch and roll attitude errors are determined by the earth sensor; yaw attitude is normally determined either by one or more gyros or by the fine sun sensor. Control is maintained by the reaction control system (thrusters). During earth mode the solar array drives can be ground commanded into the "clock" mode to track the sun as the body of the spacecraft rotates once per day about the - Y-axis to keep pointing at the earth's nadir.

Transition to normal mode operation is established by ground commands which spin up two reaction wheels and cycle the normal mode processing registers (Reference 2).

Normal and MA Mode Attitude Control

The normal attitude control mode is referred to as a short loop because all signals are processed onboard the TDRS. Earth sensor data, wheel speeds, and other sensor data are also reported to the ground in the telemetry. With inoperative earth sensors a new mode must be created that will use the MA phase information that is sent to the ground as part of a normal MA service. A fixed ground S-band, PN spread signal source at the MA frequency is required for the phase measurements. The MA calibration emitter is the signal of choice. The advantage of using the calibration emitter rather than a separate dedicated source is that the corrections generated by the MA system for its internal calibration provide the information needed to calculate TDRS body attitude changes from nominal. The 30 phases from each of the antenna elements can be processed into roll and pitch angles in well under one second. Based on the MA phase

determined roll and pitch, momentum wheel bias commands will be sent to the spacecraft via the normal S-band command link. Propagation time from TDRS to earth and back to the TDRS is about 0.25 seconds. Since the RF speed of light propagation time plus the ground calculation time is on the order of magnitude of the current onboard ACS function (1 sec), we do not expect this new long loop control mode, which we will call "MA Mode", to severely impact operations.

Several factors degrade the accuracy of the MA mode compared to normal (earth sensor) mode. The ground received MA phase shifts are due to several factors:

1. change in attitude of the TDRS relative to the earth
2. change in range from the (calibration) signal source
3. thermal drift in components
4. propagation changes in the ionosphere
5. instability of phase locked oscillators

If item 1 was the only cause of phase shifts, accuracy would be superb. Item 2 affects the analysis but not the accuracy. It should be noted that the operational MA system assumes the spacecraft body is nadir pointed when it calculates the calibration phases. If the body is off-pointed, the calibration will allow the MA to continue to work, but the SGL signal will degrade.

Next will be described the tests and analysis performed to evaluate the feasibility of creating the MA attitude control mode defined above.

Feasibility Measurement and Analysis

A series of tests conducted at WSGT and STGT have investigated the RF links required for performing attitude control with the MA system. The tests to date can be divided into three sections:

1. Forward Interference Test - Will the MA forward noise signal interfere with S-band control of a TDRS?

2. Forward Beacon Test - Can the ground station receive a forward noise signal and detect the main beam as it is swept across the station? This is required in order to transition from sun mode to earth mode, normal mode or MA mode.
3. Return Phase Test - How stable are the phases received when pointing an MA return beam at a stationary emitter?

These tests are described in more detail below.

Forward Interference Test

The MA forward system in a TDRS requires a forward drive signal for standard operation. The proposed MA ACS system acquisition mode would use the forward MA system with no input signal; thus, the forward drive signal will be noise amplified by the onboard RF equipment. The main operational concern is whether or not this noisy forward signal will couple into the S-band telemetry, tracking, and control (TT&C) receive equipment on the TDRS and interfere with operation of the satellite. During normal operations with earth sensors, the TT&C is switched to Ku-band before an MA S-band emission is activated.

The interference test showed no change in the operation of the S-band command and telemetry system as a function of the operation of MA Forward system with a noise drive signal. Our conclusion was that use of the MA forward system for acquisition would not interfere with S-band TT&C.

Forward Beacon Test

The next step was to sweep the forward noise signal across the WSGT by commanding the TDRS MA forward phase shifters. The forward signal is again a noise signal from the MA forward equipment on the satellite, since there was no link with the ground station through the K-band SGL, thus simulating the

situation that would exist during an attitude reacquisition. During this test the cold sky measurement was taken as baseline for WSGT receive equipment; the MA forward beam from 12 MA elements was sequentially stepped across WSGT in East-West and North-South directions; and the received signal levels were compared to a reference signal injected into the S-band simulation receive equipment at WSGT and recorded.

Figure 1 shows the noise level when the beam was pointed directly at WSGT as well as 12° east of WSGT. When pointed 12° east, the power was less than 0.1 dB above background. Figure 2 shows the MA forward antenna pattern received at the WSGT when the noise emission was scanned along the North-South direction to simulate the spacecraft rolling prior to earth acquisition.

Return Phase Test

The Return Phase Test conducted at STGT was a first cut at determining the accuracy and resolution of the MA return system's capability of measuring the spacecraft attitude when in an MA attitude control mode. From earth sensor telemetry data we confirmed that the satellite had a stable attitude, and therefore, the phase shifts required to point the beam at the stationary user were not influenced by changes in spacecraft attitude.

The STGT Multiple Access Beamforming Equipment (MABE) can provide operators with extensive diagnostic information, in part, because of its digital design. Easy access is afforded to the computed amplitudes and phases required to calibrate the different wire lengths and component delays of the 30 signal paths from the MA antenna elements on the TDRS to the point where they are added for a user service. The 30 calibration vector phases were recorded every ten minutes over a four hour period at the common IF frequency. If there were no electronic drift factors and the spacecraft body was not rotating, the difference between any pair of calibration phases would be constant. When the TDRS body rotates the calibration phases change to

compensate. We have developed an algorithm to convert the calibration phase changes into a measure of roll and pitch.

The first set of 30 phases measured at time zero was used as a reference and was subtracted from all of the following phase sets. Within each set of 30 phases at a given time, the phase of the central geometric element was subtracted from the other 29 values to compensate for TDRS longitudinal motion. We are interested in rotational motion only. The remaining phases were then converted to be between $\pm 180^\circ$ instead of 0° to 360° . Using the wavelength of the incoming S-band signal, 0.131 m, the phases were then converted to lengths. Knowing the XY geometric position of each of the 30 MA elements on the TDRS body and using the length found from the phases as the Z value, we have 30 points in space for each data set, representing the normalized zero phase plane of the incoming RF signal. A least squares fit of a plane in three dimensional space to these points was calculated and the normal to the plane was determined. The normal or pointing vector orientation relative to the Z axis represents the body angle displacement from nadir. The pointing vector was resolved into roll and pitch angles. Figure 3 shows the variation of the roll and pitch angles over a four hour period.

Conclusion

MA forward S-band noise emissions are detectable on the ground and can be used as a coarse estimate of spacecraft roll during orbital insertion or recovery from an upset.

The four hours of return data collected indicates that the phase differences of an incoming plane wave on each of the 30 MA antenna elements can be used as a sensor to accurately determine the TDRS spacecraft roll and pitch attitude.

Earth sensor readings indicated a typical body divergence less than 0.01° while the least squares fit to the MA phases indicated a divergence of about 0.12° . Thus, based on the limited data, it appears that attitude control commands should

not be issued for apparent roll and pitch errors of less than 0.1° when operating in the MA mode. This would allow the SGL, MA, and SSA antennas to be pointed to well within their 3 dB beamwidth.

We are currently pursuing a more refined data analysis scheme that may yield more accurate attitude determination and we are exploring the possibility of improved results by considering the differential ionospheric phase shifts in the SGL carrier frequencies. A draft plan that outlines the further testing required for establishing an operational MA control procedure is being evaluated by NASA.

REFERENCE

1. *Ground Loop Control of Tracking and Data Relay Satellite Attitude Using RF-Interferometer Techniques*, 1992 Research and Technology Report, Goddard Space Flight Center, Paul Heffernan (Code 405), Robert Jenkins (Code 712.4), William Isley (EER), 1992.
2. *TDRS Spacecraft Systems Manual*, TMO254 Vol 1, TRW, revised Nov. 1985.

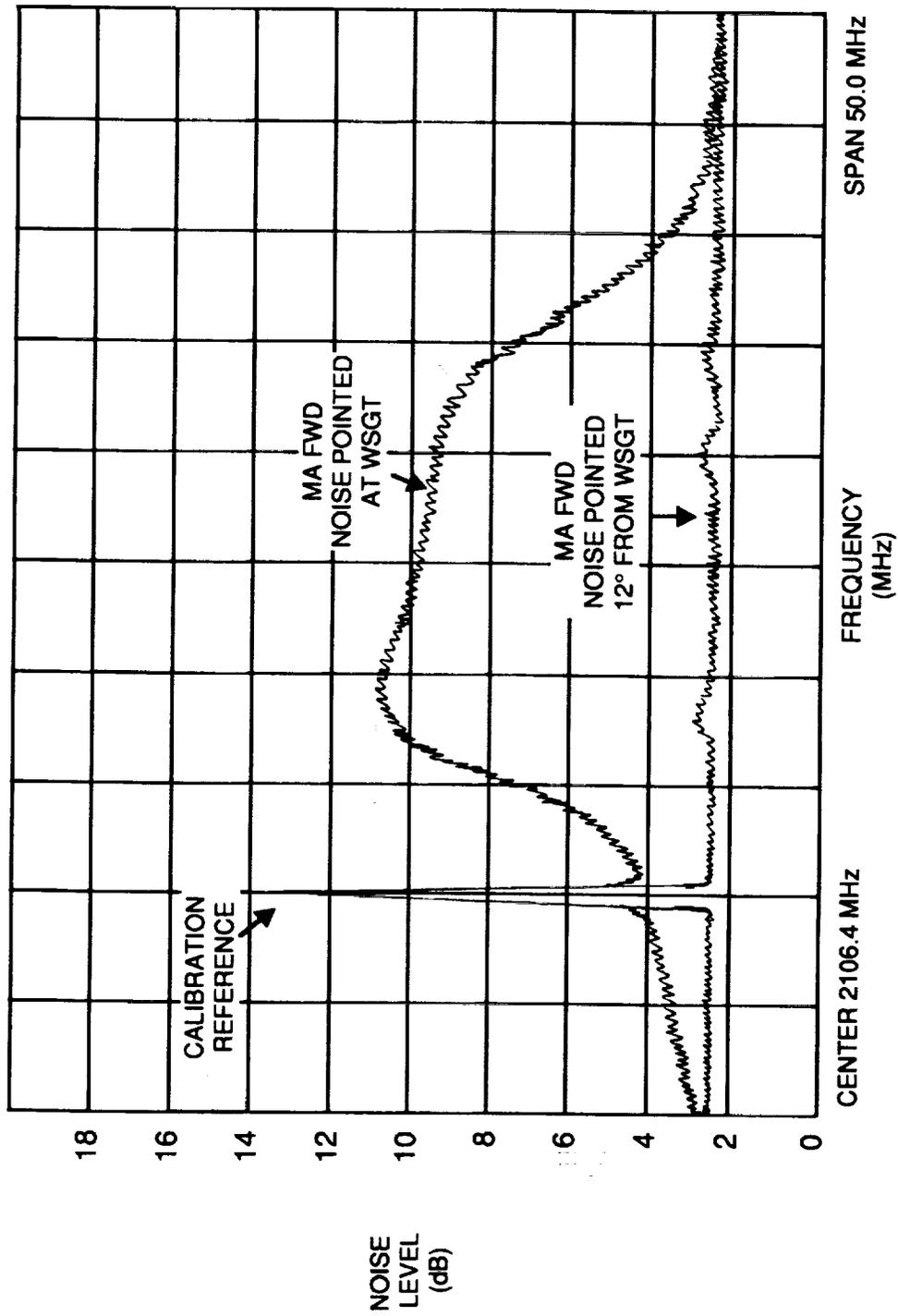


Figure 1. Forward Beacon Test

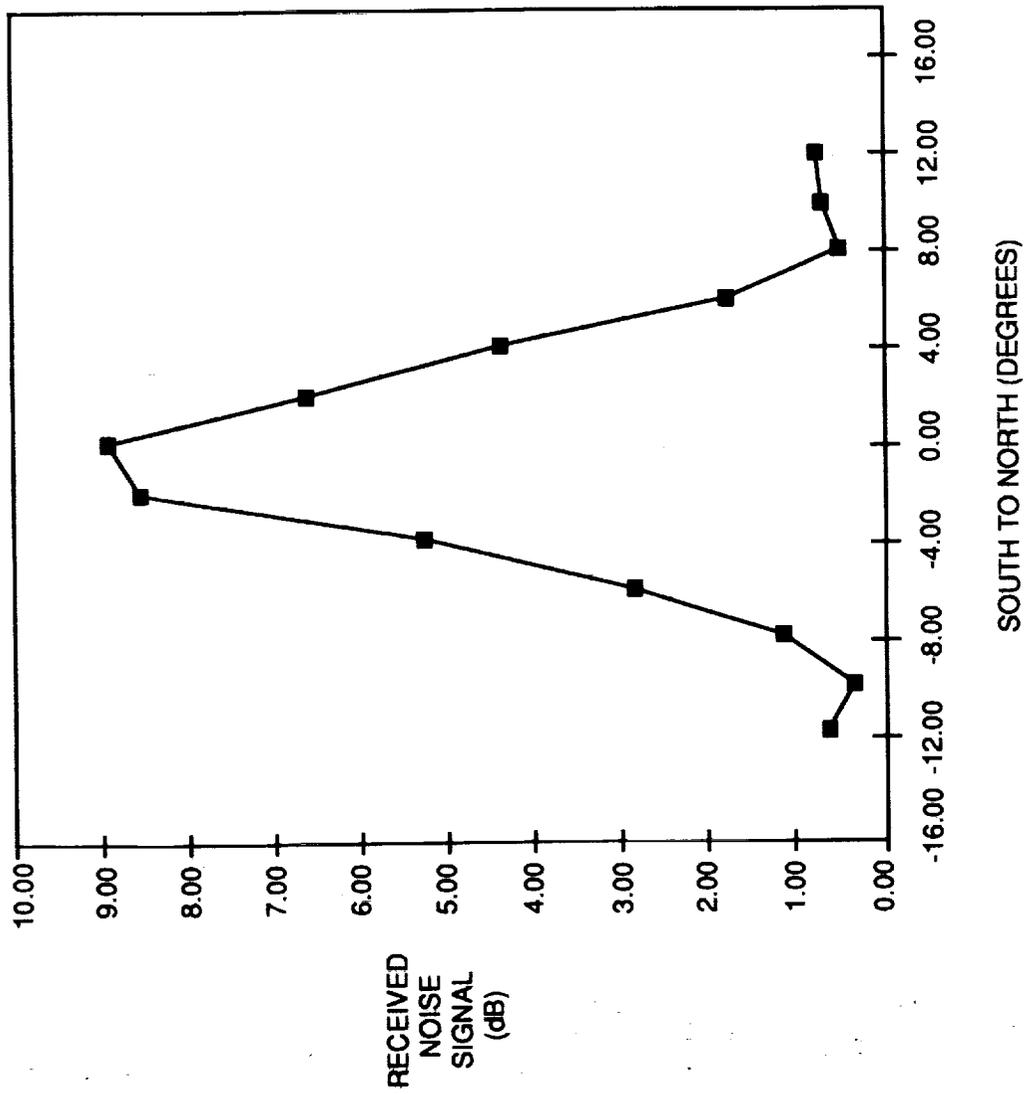


Figure 2. MA Forward Antenna Pattern

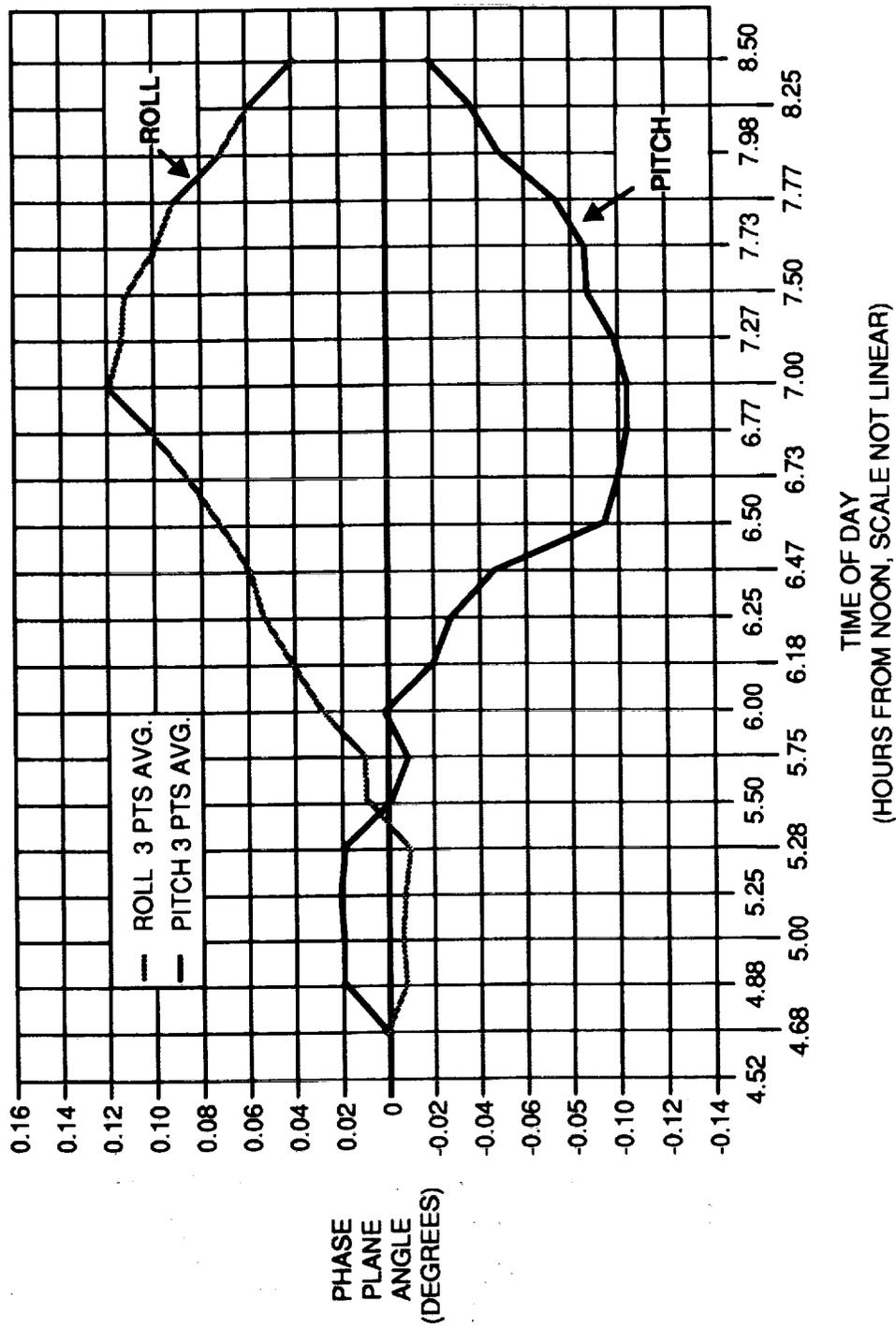


Figure 3. MA Phase Derived Attitude Stability

1. The first part of the document discusses the importance of maintaining accurate records of all transactions and activities. It emphasizes that this is crucial for ensuring transparency and accountability in the organization's operations.

2. The second part of the document outlines the various methods and tools used to collect and analyze data. It highlights the need for consistent data collection procedures and the use of advanced analytical techniques to derive meaningful insights from the data.

3. The third part of the document focuses on the implementation of data-driven decision-making processes. It provides a detailed overview of the steps involved in identifying key performance indicators (KPIs) and using data to inform strategic decisions.

4. The fourth part of the document discusses the challenges and risks associated with data management and analysis. It offers practical advice on how to mitigate these risks and ensure the integrity and security of the data.

5. The fifth part of the document concludes by summarizing the key findings and recommendations. It stresses the importance of ongoing monitoring and evaluation to ensure that the data-driven approach remains effective and relevant over time.

6. The sixth part of the document provides a detailed overview of the data collection and analysis process, including the specific steps and tools used. It also includes a list of references and a glossary of key terms.

7. The seventh part of the document discusses the future of data-driven decision-making and the role of emerging technologies. It explores the potential of artificial intelligence, machine learning, and big data in transforming organizational operations.

8. The eighth part of the document provides a final summary and a call to action. It encourages organizations to embrace a data-driven culture and to continuously invest in their data management and analysis capabilities.

**SESSION 4: MODELING AND
SIMULATION**



THE COMPOSITE ANALYTIC AND SIMULATION PACKAGE OR RFI (CASPR)
ON A CODED CHANNEL

By Jeff Freedman and Ted Berman

Stanford Telecom
7501 Forbes Blvd.
Seabrook, Md 20706
(301) 464-8900

ABSTRACT — CASPR is an analysis package which determines the performance of a coded signal in the presence of Radio Frequency Interference (RFI) and Additive White Gaussian Noise (AWGN). It can analyze a system with convolutional coding, Reed-Solomon (RS) coding, or a concatenation of the two. The signals can either be interleaved or non-interleaved. The model measures the system performance in terms of either the E_b/N_0 required to achieve a given Bit Error Rate (BER) or the BER needed for a constant E_b/N_0 .

I. INTRODUCTION

Stanford Telecom developed CASPR for NASA Goddard Space Flight Center's (GSFC) Communications Link Analysis and Simulation System (CLASS). CASPR determines the effect of RFI on a satellite communications signal transmitted across a non-linear Tracking and Data Relay Satellite II (TDRS-II) channel. This system required the development of several new and unique analytic algorithms for determining the signal performance in the presence of RFI.

CASPR determines the performance of a signal in the presence of RFI and AWGN. The performance is measured in terms of either the E_b/N_0 required to achieve a given BER or the BER needed for a constant E_b/N_0 .

Two types of RFI are analyzed: pulsed Gaussian noise and pulsed sinusoidal noise. CASPR can model multiple pulsed Gaussian and sinusoidal sources transmitting simultaneously with Poisson or periodic arrival distributions and unique EIRPs, duty cycles, and pulse durations.

CASPR supports a wide variety of coding schemes and system parameters. These are shown in Table 1.1.

Table 1.1: Input Parameters

Parameter	Possible Values and Units
E_b/N_0	any real number in dB
Data Rate	
Type of Coding	1) Convolutional rate, 1/2 or 1/3 2) RS (255,223) 3) Concatenated Convo/RS (Either may be with or without ideal interleaving)
Signal Format	NRZ, BiPhase
PN Coding	Yes, No
Signal Modulation	BPSK, QPSK
Clip Level (AM/AM)	dB's above mean signal power
AM/PM	dB/deg above mean signal power
RFI	See Section II

SECTION II. RFI ENVIRONMENT

CASPR is designed to predict the performance of TDRS user satellite communications in the presence of RFI and AWGN. As shown in Figure 2.1, RFI can corrupt communications signals for both forward and return links, primarily corrupting the space-to-space links.

CASPR treats RFI pulses with common characteristics (pulse duration, type, EIRP, and duty cycle) as a single source. The pulse duration is the length of time the pulse exists; the type is either Gaussian or sinusoidally varying (SV); the EIRP is the RFI power; and the duty cycle is the proportion of time the pulse is present. Noise-like RFI is modeled as pulsed bandlimited additive white Gaussian noise (AWGN). SV RFI is modeled as pulsed sinewave interference with a constant amplitude and a random initial phase and frequency. The frequency of the sinusoidal

This work was supported by NASA Goddard Space Flight Center under contract NAS 5-31500.

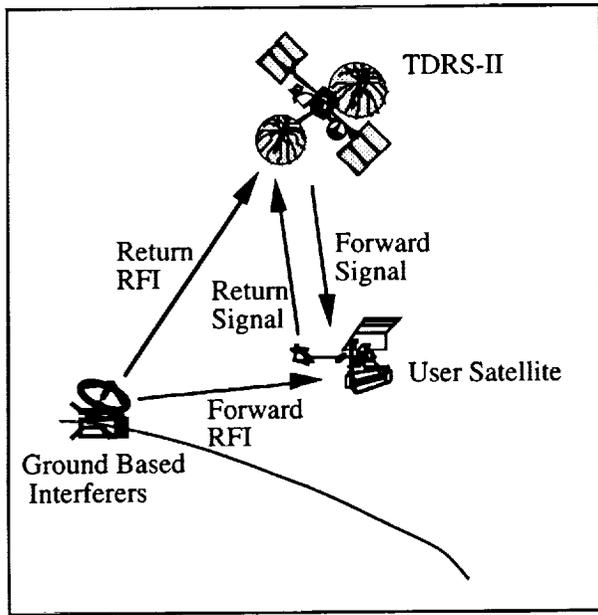


Figure 2.1: RFI Noise in Satellite Communications

pulse is constant over the pulse duration. However, different pulses have different frequencies, which are uniformly distributed within the TDRS transponder bandwidth.

SECTION III. OVERVIEW

This section gives an overview of the model. The system includes two subsystems: the channel subsystem and the coding subsystem. This section also describes the input/output at each stage of the model.

Figure 3.1 shows the CASPR flow diagram. The user first enters input parameters to the channel subsystem, which models the signal and the RFI environment and subsequently predicts symbol statistics at the input to the channel decoder. These symbol statistics are then entered into the coding subsystem, which determines the BER.

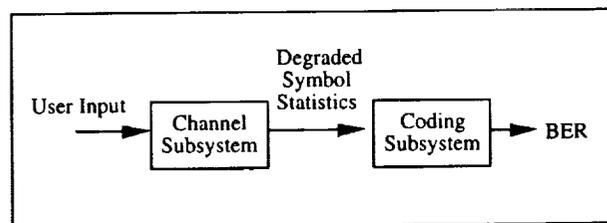


Figure 3.1: The CASPR Model Flow Diagram

The channel subsystem uses the system parameters as input and predicts symbol statistics at

the TDRS-II receiver (before coding). New algorithms were developed which use a characteristic function approach to quickly determine the symbol statistics at the TDRS-II receiver. In addition, the channel system contains a Monte-Carlo simulator that determines more precise statistics. A single analysis can use either the Monte-Carlo simulator or the characteristic function approach to determine the symbol statistics.

The coding subsystem uses the symbol statistics generated by the channel subsystem to determine the signal BER. The symbol statistics characterize the symbols at the input to the decoder. The symbol statistics for each source include the average burst length and the 8-level soft decision bin probabilities of the symbols. These bin probabilities correspond to the probability of a received symbol being quantized to a particular soft decision level of the Viterbi decoder.

The RFI, which occurs in pulses, causes burst errors in the signal. Each RFI source causes a different type of burst statistic. The analysis separately characterizes the signal degradation due to each of the RFI sources and for AWGN alone. The analysis also calculates the probability of transitioning from one type of source to another, based upon the duty cycles and characterizes the signal degradation for the system as a whole. This conditional information is used in the coding subsystem modules discussed below.

The system uses one of several different program modules, depending on the coding scheme and whether or not the symbols are interleaved [1,2,3]. CASPR uses a new importance sampling algorithm to determine the BER for a non-interleaved, convolutionally encoded signal [1]. It uses a new Markov chain analysis to determine the BER for a non-interleaved RS coded channel [2], and it uses a new combined simulation and Markov chain analysis to determine the BER for a concatenated convolutional / RS encoded signal [3].

IV. CHANNEL SUBSYSTEM

Two different modules exist in this model which can calculate the degradation due to AWGN and RFI noise. The first is an analytic model which uses a characteristic function approach, and the second is a simulation.

The analytic model has a run-time which is independent of the BER, whereas the simulation's run-time is inversely proportional to the BER. The analytic model is recommended when a low BER is required (i.e. less than 10^{-4} before any coding gain is taken into account). The simulation model is slower in many cases, but makes fewer assumptions.

A. The Characteristic Function Module

The analytic model uses a characteristic function approach. The characteristic function approach is based on the general approach described in [4] but uses new characteristic function equations. The analysis separately calculates the characteristic function of a desired symbol and of each RFI source which interferes with the desired symbol. It then combines these characteristic functions by multiplying them together. Finally, it calculates the 8-level soft decision bin probabilities from the characteristic function.

The model has two modes of operation, depending on the pulse width of the RFI. If the RFI pulse width is larger than the symbol, the errors will occur in bursts. In this case, the model separately calculates symbols corrupted with each RFI source. If, on the other hand, the RFI pulse width is smaller than the symbol duration, the module calculates a composite characteristic function.

B. The Simulation Module

The simulation module calculates the degradation due to AWGN, RFI, and amplifier nonlinearities through the use of a straightforward simulation. Additionally, it demodulates and detects the distorted symbols. It then 8-level quantizes the symbols and tabulates the statistics discussed in Section I.

V. CODING SUBSYSTEM

The coding section of the analysis follows the symbol statistics section. Its inputs are the outputs from the symbol statistics section. In summary, these are the number of RFI sources, their lengths, the probability of transitioning from one type of source to another, and a set of 8-level soft decision levels for each source.

This subsystem can analyze systems which use either convolutional coding (with Viterbi decoding), RS coding, or a concatenation of the two. It can analyze any of the above systems both with and without ideal interleaving. This interleaving is on the bit level for the convolutional code and on the symbol level for the RS code. For the concatenated coding system, this interleaving may be on either or on both codes. A list of all possible analyses follows:

1. No coding
2. Convolutional coding only
 - 2a. With convolutional interleaving
 - 2b. Without convolutional interleaving
3. RS coding only
 - 3a. With RS interleaving
 - 3b. Without RS interleaving
4. Concatenated Convolutional / RS coding
 - 4a. With both convolutional and RS interleaving
 - 4b. With convolutional interleaving only
 - 4c. With RS interleaving only
 - 4d. Without any interleaving

Due to the large number of possible analyses, it was necessary to develop several different algorithms, including both simulation and analytic models. The total number of algorithms incorporated here is six, three of which have been specially developed for this model. The following is a list of these six, the last three of which are the new models.

1. Uncoded (UnCod)
2. R_0 analytic approximation (R_0) [4]
3. Conventional convolutional coding simulation (VitS)
4. Importance sampling convolutional coding (ImpS) [1]
5. RS stand-alone analytic, using a Markov Chain (MC) [2]
 - 5a. With interleaving (RS-MC (A))
 - 5b. Without interleaving (RS-MC (B))
6. RS concatenated analytic, using a Markov Chain [3]
 - 6a. With interleaving (Con-MC (A))
 - 6b. Without interleaving (Con-MC (B))

Figures 5.1 - 5.4 show the relationship between the type of analysis run and the model which is used. Also given are the type of bursts which occur at each step. The letter F stands for fixed, G stands for geometrically distributed, B stands for bursts, and W stands for the wait time in-between bursts. For example, FB means that the bursts have a fixed length.

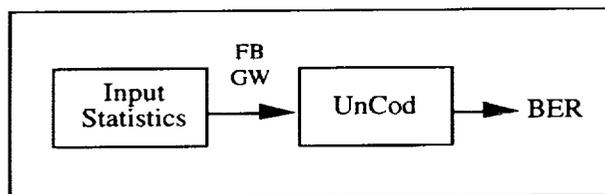


Figure 5.1: No Coding

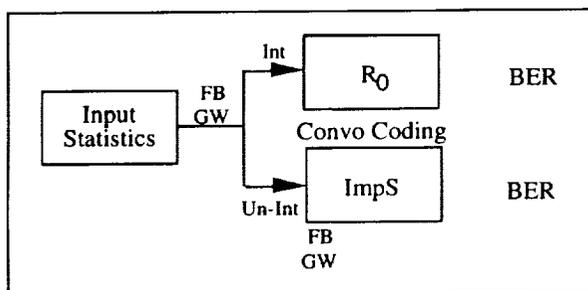


Figure 5.2: Convolutional Coding Only

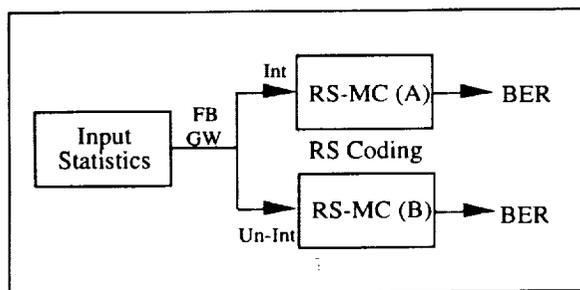


Figure 5.3: RS Coding Only

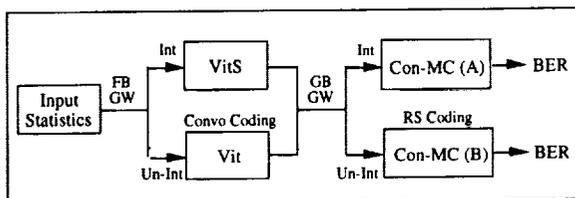


Figure 5.4: Concatenated Convolutional/RS Coding

These models contain two different types of burst statistics. Note that the input to the first level of coding in each case has bursts which are of a fixed length, but which have a geometrically

distributed wait time in-between them. This is taken directly from the input statistics of the model. However, in the concatenated model, the input statistics to the RS coder have burst lengths which are geometrically distributed. The reason for this is discussed below, in Section C.

We outline the algorithms which the coding subsystem uses in each of the six models in the following paragraphs. Run-times given are for an HP 9000 series 800 computer.

A. The Uncoded Module (UnCod)

This module simply adds up the noisiest four of the eight soft decision levels to come up with the BER.

B. The R_0 Analytic Approximation Module (R_0)

This module first determines the cutoff rate, R_0 , from the 8-level soft decision values for the channel as a whole, using a simple equation. It then determines the BER through a curve fit which is based upon a simulation. The total run time for this module is less than one second.

C. The Conventional Convolutional Coding Simulation Module (VitS)

The VitS module is a straightforward simulation. The noisy channel outputs are simulated using a random number generator. The decoding process is also simulated in a way which directly corresponds to the decoding process of an actual Viterbi decoder. The output statistics here are the BER, the mean burst length, and the mean time between bursts.

Both the burst length and the time between bursts are assumed to be geometrically distributed, where a geometric distribution is defined as follows:

$$P(b = m) = P_B(1 - P_B)^{m-1}, \quad m > 0$$

where $P_B = 1 / \text{mean burst length}$

Note that the probability of burst length or wait time falls exponentially as the length or the time increases. The model assumes that half the bits in a burst are in error, including the first and last bit. By definition, none of the bits in a wait time are in error.

Since this module is a simulation, its run-time is inversely proportional to the output BER. The Viterbi decoding process is relatively complicated, so the run-time for a low BER could be prohibitively long. However, for concatenated convolutional / RS codes, the necessary output BER of the Viterbi decoder is relatively high, even if the total system BER is very small. Typically, an inner code BER of 10^{-3} is sufficiently small. This BER can be simulated with reasonable accuracy in about 5 minutes.

D. The Importance Sampling Convolutional Coding Module (VitS)

The ImpS module has the same inputs as the VitS module, and it uses the same geometric distribution assumption. The only output, however, is the BER.

Since this module may have to deal with very small BER's, it must be more efficient than the VitS module. It achieves this increase through importance sampling, a technique used to bias the channel statistics so that error (important) events occur more frequently than they would in an unbiased simulation [5]. It uses a weighting function to offset this bias. This function gives the ratio of the probability of a specific error occurring in an unbiased channel to the same error occurring in the biased channel. These values are computed through the use of a Markov chain technique.

The run-time of this model is independent of the output BER. To get a reasonable level of accuracy takes about 1 hour with this model. Note that for a BER of less than about 10^{-4} , the run-time would be less if the VitS model were used instead of the ImpS model. Therefore, the model always initially attempts to obtain the BER through the VitS module. However, if it appears that the BER will be less than 10^{-4} after a short period of time (100,000 bits), the model switches over to the ImpS module.

E. The RS Stand-Alone Analytic Module (RS-MC)

The RS-MC analytic module makes several simplifications in order to make the analysis achievable. First, each bit may be in only one of two states: the burst state or the no-burst state.

The statistics of the no-burst state are found by taking the weighted average of the statistics of all the input states except the no-burst state. Second, each burst is assumed to completely overlap a symbol or symbols, so that either all of the bits in a symbol or none of the bits in a symbol are in the burst state.

The analysis proceeds as follows: the module uses the input statistics to calculate a probability of symbol error for each of the two states. A MC calculates the probability that a symbol is or is not hit by a burst. If a symbol is hit, the MC calculates the probability that the symbol is at the beginning, the end, or anywhere in the middle of the burst. If RS interleaving is used, RS-MC (A) calculates the BER based upon symbol errors which are independent of one another. If RS interleaving is not used, RS-MC (B) calculates the BER based upon dependent symbol errors (several symbols in a row in error).

This is an analytic module, so the run-time is very short, about 5 seconds, and is independent of the BER.

F. The RS Concatenated Analytic Module (Con-MC)

The Con-MC analytic module assumes that the bursts which are output from the Viterbi decoder and into the RS encoder come in bursts which have geometrically distributed burst and wait times. The mean value of these two distributions (the only value necessary to characterize them) is determined in the VitS module described above.

This model uses two Markov chains. The first calculates transition probabilities for transitions from one bit to another. The second calculates transition probabilities for transitions from one symbol to another. Module Con-MC (A) calculates the BER for the ideal interleaving case. It uses information from the first MC only and assumes that bursts errors occur for bits within a symbol. However, it assumes that these bursts errors do not occur from one symbol to another, so symbol errors are independent. Module Con-MC (B) calculates the BER for the non-interleaved case and uses information from both Markov chains. It assumes a geometric distribu-

tion of burst errors throughout, so symbol errors occur in bursts.

Like the previous module, this one is analytic, so the run-time is very short, about 5 seconds, and is independent of the BER.

VI. RESULTS

This section presents results for codes generated by three different coding systems: a convolutional code, a RS code, and a concatenated convolutional / RS code. These correspond to Models 4, 5, and 6, respectively. Each figure graphs the BER against E_b/N_0 . The convolutional codes used are rate 1/2, constraint length 7. The RS codes used are (255,223) codes, which have 8 bits per symbol.

Figure 6.1 shows results for convolutional coding, both with and without interleaving, and for an RFI environment with six sources. Each

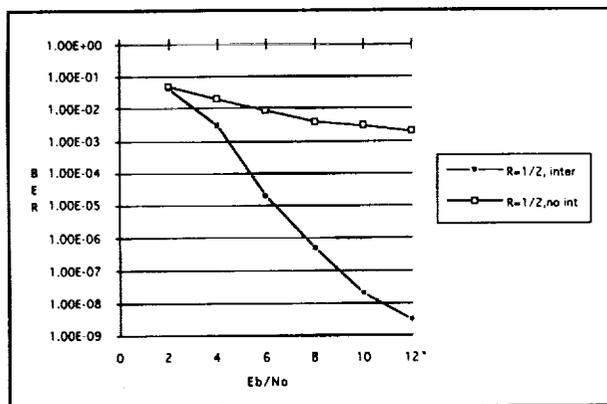


Figure 6.1: Convolutional Coding With and Without Interleaving

RFI burst has a length of 22 coded symbols. Note that interleaving results in a large improvement.

Figure 6.2 is for a RS code, again with and without interleaving with one RFI source. It has infinite power and a duty cycle of .005. We present results for bursts of lengths 5 and 20. Note that interleaving results in little improvement for a burst length of 5, but a lot of improvement for a burst length of 20.

Figure 6.3 shows results for a concatenated code without interleaving. There is no RFI here. For high BER's, the model results were compared to simulation results, and the two models agreed to within .1 dB.

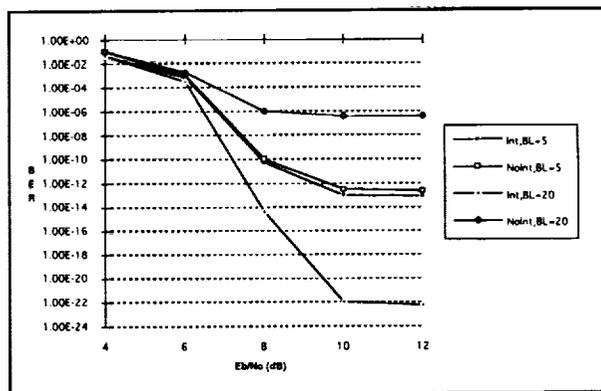


Figure 6.2: RS Code With and Without Interleaving

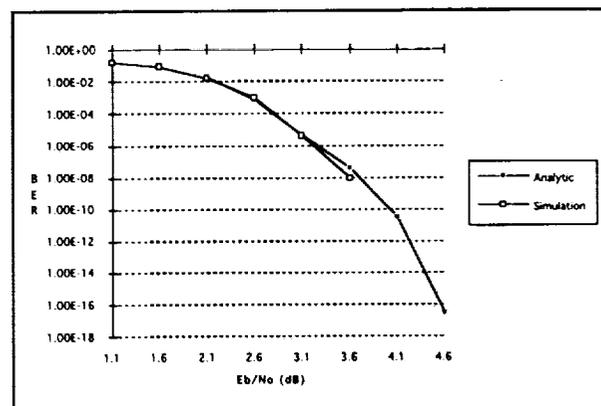


Figure 6.3: Results for a Concatenated Code Without Interleaving

VII. CONCLUSION

This paper describes a model which can analyze the degradation due to RFI for a coded communication system. It can model many coding schemes including: convolutional, RS, and a concatenated RS convolutional. It can also model any combination of interleaving or non-interleaving. The model is made up of several different modules, including four which use algorithms developed specifically for CASPR.

REFERENCES

- [1] Berman, T., and Freedman, J., "An Importance Sampling Analysis of a Non-Interleaved Viterbi Decoder in an RFI Environment," Submitted to IEEE Trans. Comm. in Feb. 1992.

[2] Berman, T., and Freedman, J., "Non-Interleaved Reed-Solomon Coding Over a Bursty Channel," MILCOM Conference Proceedings, Nov. 1992.

[3] Berman, T., and Freedman, J., "An Analytic Analysis of a Concatenated Reed-Solomon (RS) / Viterbi Coding System Both With and Without RS Interleaving," IPCCC Conference Proceedings, April 1992.

[4] Weinberg, A., "The Impact of Pulsed RFI on the Coded BER Performance of the Nonlinear Satellite Communication Channel," IEEE Trans. Comm., vol. COM-29, pp. 605-20, May, 1981.

[5] Sadowsky, J., "A New Method for Viterbi Decoder Simulation Using Importance Sampling," IEEE Trans. Comm., vol. 38, pp. 1341-51, Sept. 1990.

AN IN-DEPTH ANALYSIS AND MODELLING OF THE SHUTTLE TO MILA S-BAND TELEMETRY LINK

Armen Caroglanian
Fernando A. Pellerano
NASA Goddard Space Flight Center
Telecommunication Systems Branch
Greenbelt, MD 20771

Dale D. Shama
Loral Aerosys
Beltsville, MD 20705

Summary- The S-Band radio frequency (RF) link between the Merritt Island (MILA) Tracking Station and the Space Shuttle launch pads is a critical communication path for pre-launch and launch operations. The proposed siting of the Center for Space Education (CSE) at the Visitor Center required a study to avoid RF line-of-sight blockage and reflection paths. The study revealed the trees near MILA's 9-meter (9-M) antennas are obstructing the optical line-of-sight. The studies found diffraction is the main propagation mechanism. This paper describes a link model based on the Geometric Theory of Diffraction.

1.0 Introduction

The S-Band radio frequency (RF) link between the Merritt Island (MILA) Tracking Station at the Kennedy Space Center (KSC) and the Space Shuttle launch pads is a critical communication path for pre-launch and launch operations. The Visitor's Information Center (VIC) is located near the lines-of-sight. As new buildings are added to the VIC, care has been taken to avoid RF line-of-sight blockage and reflection paths. The proposed siting of the Center for Space Education (CSE) at the VIC required extensive theoretical and experimental studies [1], [2], [3].

The work, performed by NASA, Bendix, Lincom and ECAC, identified scattering from buildings, vegetation, and the launch pad as probable causes of link degradation. In addition, weather was found to be a factor in altering the RF propagation. Field probes in front of the MILA 9-M antennas determined the trees in the near-field were placing a severe illumination taper on the antenna aperture. The modeling of the RF perturbation caused by the trees is the focus of this report.

2.0 RF measurements

Currently, an optical line-of-sight does not exist between the 9-M antennas and the launch pads. Maximum signal strength at MILA is achieved when the 9-M antennas are pointed at the top edge of the forest. The maximum signal pointing angle is above the line-of-sight angle, meaning the top edge of the trees appears to be a diffraction source. Previous studies [4] have modelled this effect as knife-edge diffraction. Multipath, however, has also been detected. Improved test procedures to measure the influence of diffraction and multipath individually were conducted during a test trip to MILA in May 1992. Data analysis showed that, although multipath was detected, diffraction was a stronger mechanism [5]. A theoretical forest attenuation model developed George Washington University was also employed [6]. The model determined the forest was "hard". That is, the average attenuation is high, therefore multipath effects will be minimal. Using the measurements and the model, it was determined that link maintenance analysis would be based on diffraction models.

3.0 Diffraction Theory and Modelling

The MILA diffraction model is based on the Geometric Theory of Diffraction [7]. The model has two modes of operation. One mode calculates RF energy incident at a given point on an antenna's surface in the presence of blockage. The second mode calculates the total received RF power of a reflector antenna due to either direct (free-space), diffracted, or the combined RF energy.

The RF energy originates from an omnidirectional power source at the launch pad and may be partially blocked by a perfectly-conducting knife-edge obstacle (see Figure 1). This set-up closely parallels the actual conditions at MILA. In the MILA link, the polarization of the transmitted signal is circular. The MILA diffraction model, however, utilizes linear polarization. Circular polarization can be modeled by properly phasing and summing the two orthogonal linear received electric fields.

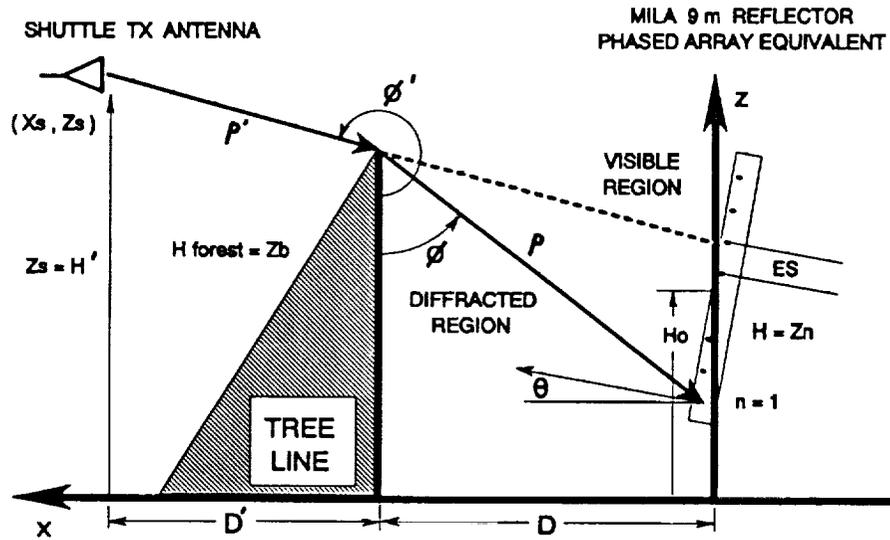


Figure 1 MILA diffraction model geometry.

The received electric field at the output of the 9-M antenna may be calculated on the basis of the following development. The incident (i.e. non integrated) direct E field is found as,

$$E_o(h) = Z(h) \frac{\lambda}{4\pi R} e^{-jkR} \quad (1)$$

where h = vertical height on the 9-M antenna aperture

R = distance from the transmit antenna to the 9-M antenna

k = the wavenumber = $2\pi/\lambda$

$Z(h) = 0$, blocked

$= 1$, unblocked

Function $Z(h)$ determines, by the given geometry, whether the transmitted energy is blocked by the tree-line or not. The incident diffracted E field at the 9-M antenna is found as,

$$E_d(h) = \frac{\lambda}{4\pi\rho'} * e^{-jk\rho'} * CDCH * A e^{-jk\rho} \quad (2)$$

where ρ' = distance from the transmit antenna to diffraction edge

ρ = distance from diffraction edge to the 9-M antenna aperture

CDCH = diffraction coefficient for vertical polarization

(or CDCS = diffraction coefficient for horizontal polarization)

A = spatial attenuation function for diffracted power.

The diffraction coefficient CDCH, a complex quantity, is generated from an expression dependant upon the incidence angle ϕ' , the reflected angle of diffraction ϕ , the distance from the diffracting edge to the source, D' , the distance from the diffracting edge to the receive antenna, D , and the assumption that the diffracting edge is a perfectly-conducting half-plane. The function A, represents the spatial attenuation for the diffracted field, and is given by,

$$A = \sqrt{\frac{\rho'}{\rho(\rho'+\rho)}} \quad (3)$$

The received E field is found as a function of the incident signal integrated over the gain function of the receive antenna aperture. The model assumes no E field variation in the horizontal direction of the aperture. This is a valid assumption because the direct and diffracted fields incident on the antenna are planar and cylindrical fields, respectively. The gain function of the receive antenna is generated by a separate reflector antenna model [8]. The model breaks up the aperture into horizontal segments of width $ES \lambda$ and calculates the complex values for the electric field (far field) on the aperture. An equivalent field for the segment is generated, thus the 9-M antenna gain model becomes a phased array with vertical dependance only. This simplification greatly reduces the computation time for the model. For the present study, an optimal value of $ES=1$ was used. The antenna may also be pointed at an angle θ with respect to the horizontal plane. The received (integrated) field, E_r , can be found as,

$$E_r(H_{forest}, D) = \sum_k G_k(h) [E_{o_k} + E_{d_k}] \quad (4)$$

where H_{forest} = height of the forest edge

D = distance of the forest edge to the 9-M antenna

$G(h)$ = gain function of the 9-M antenna

E_{o_k} = incident direct field (equation 1)

E_{d_k} = incident diffracted field (equation 2)

It is mathematically possible to analyze the contributions of the direct and diffracted fields independently. All cases can be analyzed as a function of blockage height and distance from the antenna, signal frequency, or antenna pointing angle. Both the magnitude and phase relationships are calculated for each case and can be plotted in terms of electric field or power.

4.0 Simulation Results and Analysis

The first check of the GTD software was the calculation of the 9-M antenna receive elevation pattern. Figure 2 shows the comparison of the experimental and theoretical results. The peak receive signal, as expected, occurs when the 9-M antenna is pointed to an elevation angle corresponding to the closest tree-line. The experimental and calculated patterns show good agreement out to the first sidelobe. The slightly higher than expected receive energy at low elevation angles may be due to forest multipath and/or ground reflections.

Figure 3 shows the MILA 9-M received energy for a simulated forest being trimmed in height. The 9-M received energy contributions for direct, diffracted, and total signal are shown. For the current blockage case, the trees near the 9-M are at a height of 45 ft. The trees would have to be reduced in height to 30 feet, all the way out to the VIC, to recover the direct signal. Figure 4 shows the results of trimming the forest back and leaving the closest trees at a height of 45 feet. The trees would have to be trimmed back nearly 5,000 feet for

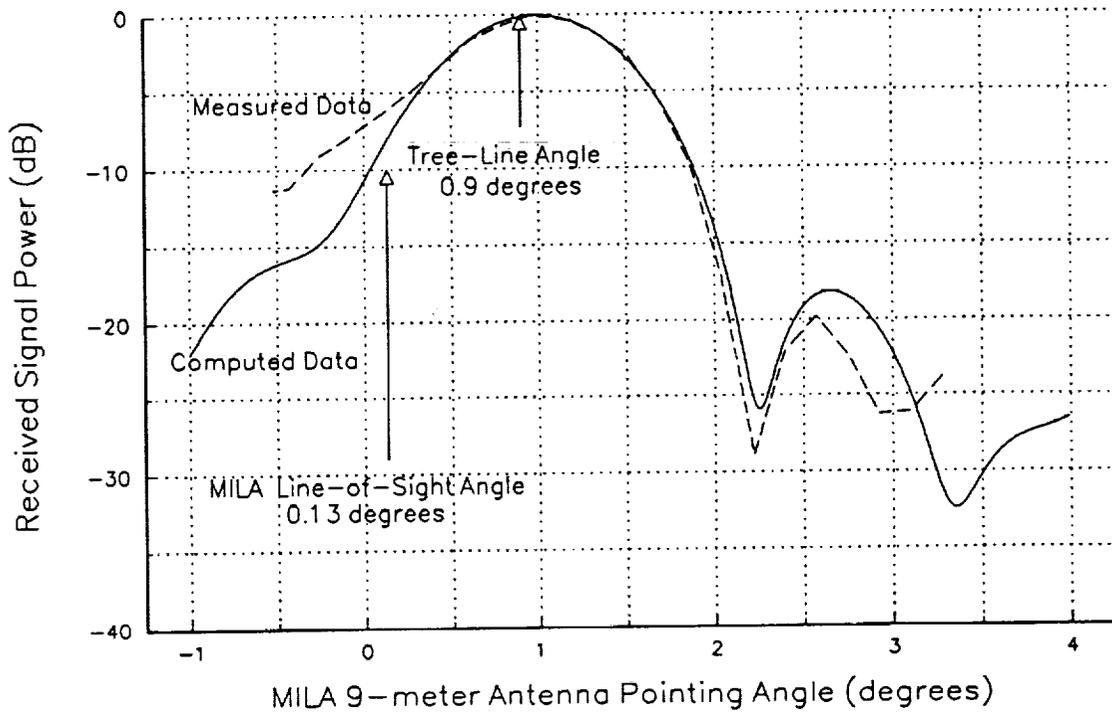


Figure 2 MILA 9-M antenna pattern.

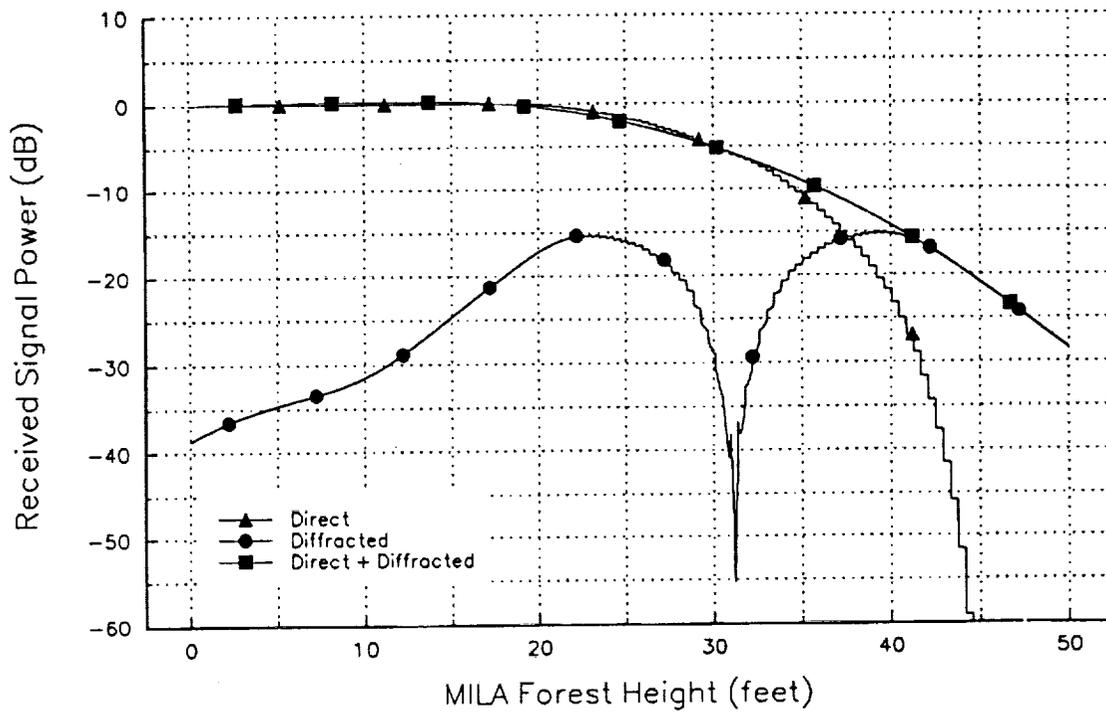


Figure 3 MILA 9-M received power vs. forest height

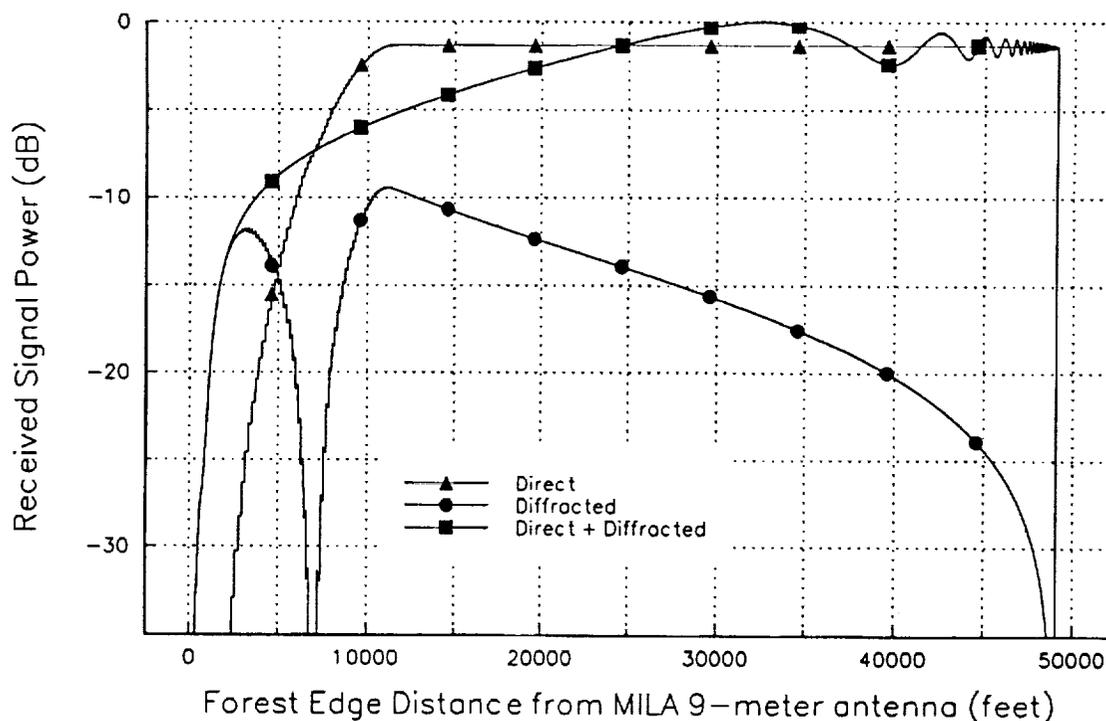


Figure 4 MILA 9-M received power vs. forest edge distance.

the direct signal to match the diffracted signal level. This approach would result in a mile long deforested path.

5.0 Forest Characterization

To apply the model to the MILA forest, the forest height and tree locations are needed. The bulk of the trees are located along the swampy mile-long path between MILA and the VIC. During the MILA link studies, surveyors targeted selected tree-lines and obtained a few height/position measurements.

A technique known as stereoscopic photography is being used to map out the heights and positions of the trees. Photos are taken of the trees from the vantage point of a cherry picker located at MILA. Two pictures, separated 50 to 100 feet, are taken at precisely known locations. The photos are then scanned into a computer. The location of the Vehicle Assembly Building and rockets at the VIC are used as reference objects for the analysis. The

tree image position and image shift between the right and left photos is used by a spreadsheet program to calculate the tree's height and position on map.

6.0 Conclusions

The GTD Diffraction Model is a valuable tool for modelling the MILA/Shuttle S-Band RF Link. Tree cutting at MILA is extremely difficult due to strict environmental regulations. The diffraction model is currently being used to evaluate various tree cutting scenarios at MILA. The preferred approach is selective cutting. With the help of the GTD Diffraction Model, the benefits of a potential tree cutting can be evaluated and justified.

7.0 References

- [1] W.C. Peng, "CLASS MILA Interference Analysis", NASA/GSFC Technical Report TR-8707-03, 1991.
- [2] A. Caroglanian, editor, "MILA/Pad 39 S-Band Shuttle Support Link RF Testing and Launch Monitoring Study Report", NASA/GSFC Technical Memorandum, 1991.
- [3] D.Kenney, et al, "Analysis of MILA 9M Antenna to Orbiter S-Band PM Subsystem", ECAC Engineering Memorandum, EM-91-DN-001.
- [4] R.F.White, et al, "Engineering Considerations for Microwave Communications Systems", GTE Lenkurt, 1970, p.35.
- [5] F.A.Pellerano, "A Study of the Effects of Vegetation on the Performance of S-Band Radio Frequency Links", NASA/GSFC Professional Intern Program Report, May 1991.
- [6] S.S. Seker, "Determination of an Electromagnetic Model for Vegetation in Support of the MILA/Shuttle RF Link Study", GWU Technical Report, 1991.
- [7] C.A.Balanis, "Advanced Engineering Electromagnetics", Wiley & Sons Publishers, New York, 1989.
- [8] D.D.Shama, "MILA Diffraction Model", NASA/GSFC Internal Memorandum, March 1993

THE AUTOMATED CONFLICT RESOLUTION SYSTEM (ACRS)

By Ted Kaplan, Andrew Musliner, David Wampler

Stanford Telecom
7501 Forbes Blvd
Seabrook, MD 20706
(301) 464-8900

ABSTRACT—The Automated Conflict Resolution System (ACRS) is a mission-current scheduling aid that predicts periods of mutual interference when two or more orbiting spacecraft are scheduled to communicate with the same Tracking and Data Relay Satellite (TDRS) at the same time. The mutual interference predicted has the potential to degrade or prevent communications. Thus the ACRS system is a useful tool for aiding in the scheduling of Space Network (SN) communications.

I. INTRODUCTION

NASA's Network Control Center (NCC) schedules communications of orbiting spacecraft through the Tracking and Data Relay Satellite System (TDRSS). Since most TDRSS users operate with the same polarization and at the same or similar frequencies, there is a potential for communications "mutual interference" between users. Also known as self-interference, mutual interference has the potential to occur when two or more spacecraft are communicating with the same TDRS at the same time. Naturally, as the number of concurrently orbiting user spacecraft increases, so does the probability for mutual interference.

When mutual interference occurs, it has the potential to delay signal acquisition, cause data degradation, and even loss of lock for a user spacecraft. Thus it is critical, especially during manned flight missions, for the communications schedule to avoid periods of potential mutual interference. In order to mitigate the interference, the NCC created a requirement for a Network tool that predicts mutual interference.

Stanford Telecom developed ACRS to support satellite communications scheduling by NASA's NCC. ACRS is a mission-current tool designed to analyze communications problems arising when two or more orbiting spacecraft are scheduled to communicate with the same TDRS at the

same time. ACRS provides an exhaustive in-depth look at the mutual interference potential between the many user spacecraft missions currently in operation. The detailed output charts produced, along with the suggested interference mitigation techniques, provide the NCC with an accurate tool for mission-current user scheduling and mutual interference prediction and mitigation.

The NCC uses ACRS outputs on a weekly basis to predict potential spacecraft mutual interference and on a daily basis during critical Shuttle mission support. A newly developed implementation of ACRS is currently undergoing operational prototype testing within the NCC Operations Control Room (OCR). This classified version of the software incorporates the actual NCC communications schedule and will be utilized in an ongoing fashion by the NCC operators to accurately predict potential interference periods during scheduling operations and mission planning. It is the analysis contained in this NCC OCR version of ACRS that is detailed in this paper.

II. THE ACRS SYSTEM

ACRS, as implemented within the NCC OCR, runs on an HP 9000 735 Unix workstation. The workstation is connected, via an eavesdropping LAN Probe, to the NCC Inter-Segment Network (ISN). This network connects the Flight Dynamics Facility to the NCC and carries all relevant SN schedule and orbital data in a real-time manner as it is generated. Using the LAN Probe, ACRS continually extracts the current orbital and schedule data and stores it in a specialized database system for later use. The data stored can then be used in the performance of ACRS analyses on past, present, or future mission data.

Specifying an ACRS analysis is done via a

user-friendly graphical user interface (GUI) designed to the specifications of the NCC operators who use the system. Figure 1 shows a typical input screen from the ACRS analysis system. Whereas the specifics of operating ACRS will not be discussed in this paper, below is a brief listing of some of the many functions available to the ACRS operator. For details on how to operate ACRS in the NCC OCR, see [3]:

- Specify and run an ACRS analysis.
- View a list of previously executed ACRS analyses.
- View the outputs from any previously executed ACRS analysis.
- Backup the ACRS databases to tape.
- Restore the ACRS databases from tape.
- Backup ACRS analyses to tape.
- Restore an ACRS analysis output from backup tape.
- View/modify the ACRS orbital and SN schedule databases.
- Validate the data within the ACRS databases.

III. ANALYSIS OVERVIEW

ACRS provides a clear picture of communications interference caused by and to other orbiting satellites, calculating all periods of compromised return link TDRS communications for all users. In predicting impeded communications, ACRS analyzes three areas of operation:

- Data degradation
- Acquisition delay, and
- Loss of bit synchronization.

In order to accommodate mission-current analysis for the NCC and Payload Operations Control Center (POCC) distribution, ACRS is designed to process a large number of *interference combinations* in a short period of time. An interference combination is comprised of a pair of user communication links, one called the *desired link*, and the other the *interfering link*. ACRS is capable of rapidly processing all possible interference combinations of desired and interfering links. Interference periods are calculated for each communication link of each operating TDRSS user space-

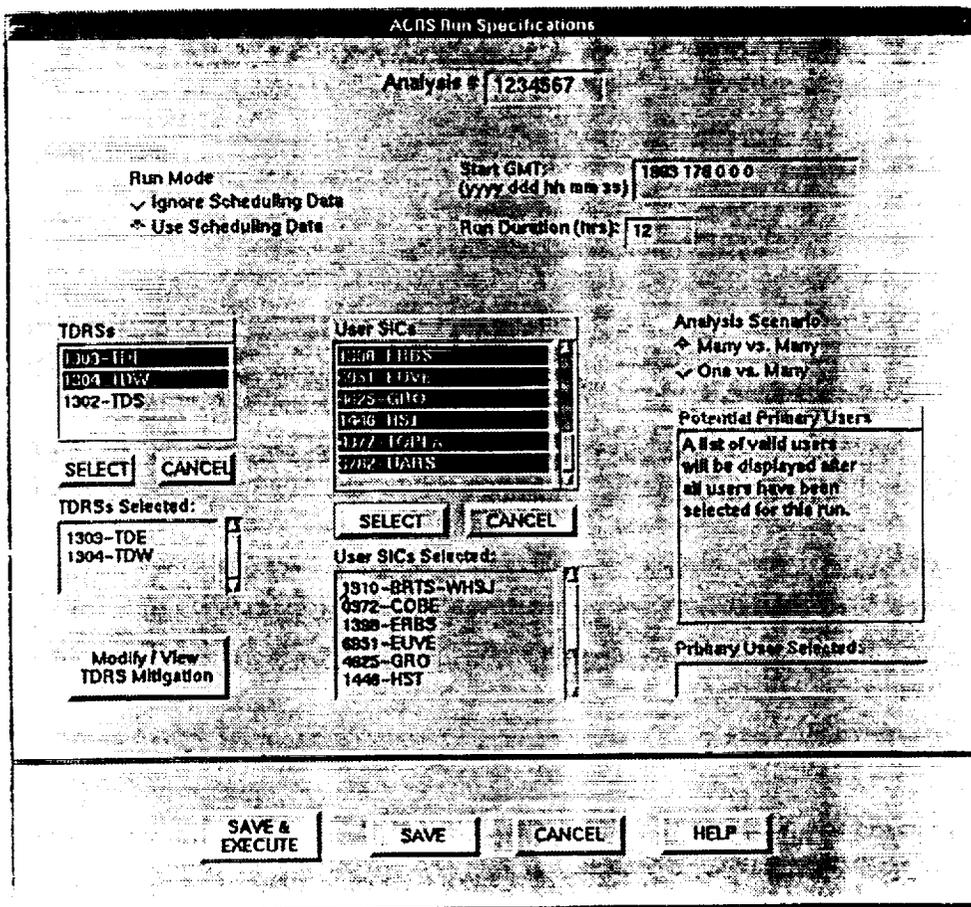


Figure 1. ACRS Analysis Screen

craft versus all other links of all other operating TDRSS user spacecraft.

The inputs for ACRS include:

- Link characteristics for all of the communication links to be analyzed.
- Accurate orbital and SN schedule information for each user.
- TDRSS receive antenna patterns.
- Analysis-specific run parameters.

The ACRS analysis involves a three-step process:

- 1) First, the required discrimination is calculated for each interference combination based on the worst case scenarios. This value indicates the necessary power differential for the uninterrupted operation of the desired link when the desired and the interfering links are in the same line of antenna boresight from TDRSS.
- 2) This value is then used to calculate the interference threshold angle which is the computed difference between the TDRS-to-interferer and TDRS-to-desired look angles.
- 3) Finally, the databased interference threshold angles for each interference combination are compared on a timepoint-by-timepoint basis with the actual inter-user angles (see Figure 2) to produce an accurate summary of possible mutual interference between user links. When the actual inter-user angle is less than the interference threshold angle, interference is likely to occur.

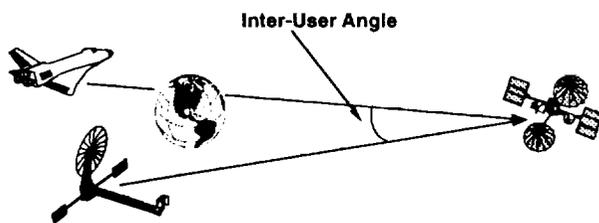


Figure 2. Inter-User Angle

Since ACRS has access (via the ISN) to the actual communications schedules, ACRS has the option to utilize these schedules in its mutual interference predictions. If the

operator selects to consider the communications schedule data, ACRS will only predict mutual interference if both the desired and interfering user are scheduled to communicate with the same TDRS at the same time (subject to the above inter-user angle criteria). Alternately, if the communications schedule data is not used, ACRS will predict mutual interference if both the desired and interfering users are simply line-of-sight closest to the same TDRS (subject to the above inter-user angle criteria). Using the schedule data has the effect of dramatically reducing the predicted interference periods.

Each of the above analysis steps is depicted in Figure 3 and described in more detail in the sections below.

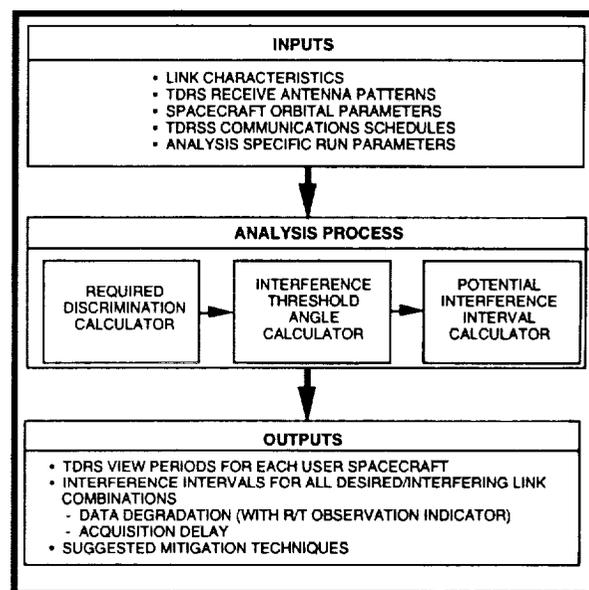


Figure 3. ACRS Analysis

IV. DETAILED ACRS ANALYSIS PROCESS

A. Calculation of Required Discrimination

The first step in the ACRS analysis entails determining the amount of discrimination required to avoid interference for each pair of desired and interfering links. This value is used as input to the Interference Threshold Angle Calculator. ACRS is presently set up for the

calculation of three types of required discrimination thresholds. These thresholds are necessary to avoid:

- 1) Data degradation
- 2) Acquisition delay, and
- 3) Loss of bit synchronization.

For threshold calculations, desired channel energy-to-noise density (E_s/N_0) or equivalently, probability of error (P_e), is determined from requirements documents and Communications Link Analysis and Simulation System (CLASS, NASA GSFC Code 531.1) analyses that correspond to operation of the desired channel with zero margin. The actual desired channel E_s/N_0 at the receiver is determined from CLASS link budgets. For each combination of links, the interfering power level, P_i/N_0 , is then found that results in the desired channel operation at the zero margin level. The difference between the actual interfering power level and P_i/N_0 is the required discrimination that would enable the desired channel to operate without interference. The actual interfering power level is determined from CLASS link budgets.

CLASS link budgets are updated regularly. The desired user's link budget includes losses due to pointing, Radio Frequency Interference (RFI), and hardware distortion that reflects the age of the TDRS satellite. Hardware distortion and pointing losses are not included in the interfering user's link budget thus resulting in a worst case interference scenario. Free space loss is calculated assuming the desired and interfering satellites are at the maximum distance from TDRS. Since the actual satellite geometries vary, the free space loss will only be accurate to within about .5 dB.

ACRS uses analytic and simulation packages to search for the interfering power level (P_i/N_0) that results in the P_e corresponding to desired channel operation with zero margin. Since hardware distortion losses are small (approximately .5 dB), the communications channel is assumed to be linear. With this assumption, a characteristic function approach can be used to find P_i/N_0 . P_e is calculated from the characteristic function of the signal with interference and noise. The program searches for the interfering power level (P_i/N_0) that results in the P_e that corresponds to

desired channel operation with zero margin.

In the case of data degradation, zero margin corresponds to $P_e = 10^{-5}$ (10^{-4} for Shuttle S-band). P_e is calculated by integrating the characteristic function of the desired signal with mixed interference signal and noise, over all phases of the interfering signal. Averaging over all phases of the interfering signal is a reasonable approximation to the actual interference situation where different Doppler rates result in a time-varying phase of the interference signal relative to the desired signal.

For acquisition delay and bit synchronization, simulation and specified values from requirements documents are used to determine the desired carrier-to-noise density that results in operation with zero-dB margin. The characteristic function approach is then used to determine the P_i/N_0 that results in desired channel operation with zero margin.

TDRSS signals are either BPSK or QPSK modulated. The I- and Q-channels of the desired signal are analyzed separately, whereas both I and Q channels of the interferer are included in the characteristic function. The characteristic function includes averaging of all phases of the interfering signal relative to the desired signal and also averaging the interfering chip polarities using Bernoulli trials. The channel noise is assumed to be Additive White Gaussian (AWG).

After simplifying, the characteristic function becomes

$$C(u) = e^{-u^2/4 + ju\sqrt{\frac{E_s}{N_0}}} \frac{1}{2\pi} \int_0^{2\pi} \cos\{u A_I \cos(\theta)\} \cos^{N_I}\{u B_I \cos(\theta)\} \sin\{u A_Q \cos(\theta)\} \sin^{N_Q}\{u B_Q \cos(\theta)\} d\theta$$

where

$$\frac{E_s}{N_0} = \text{energy-to-noise density of the desired symbol (I- or Q-channel)}$$

$$N_I = \left\langle \frac{T_S}{T_I} \right\rangle$$

The brackets indicate a floor function with the fraction rounded to the lowest integer.

T_S = the symbol duration of the desired (I- or Q-) channel.

T_I = interfering signal, I-channel, chip duration. If the desired signal is PN modulated, the interfering chip duration

is the smaller of the interferer symbol duration and the desired PN rate.

T_Q = interfering signal, Q-channel, chip dura-

$$N_Q = \left\langle \frac{T_S}{T_Q} \right\rangle$$

tion.

= effective amplitude of the I-channel chip,

$$A_I = \sqrt{\frac{E_I T_I}{N_0 T_S}} ;$$

interfering with the desired symbol.

= effective amplitude of the Q-channel chip,

$$\frac{E_I}{N_0} = \text{energy-to-noise density of an interferer I-channel chip.}$$

$$A_Q = \sqrt{\frac{E_Q T_Q}{N_0 T_S}} ;$$

interfering with the desired symbol.

= fractional part of an I-channel chip, inter-

$$\frac{E_Q}{N_0} = \text{energy-to-noise density of an interferer Q-channel chip.}$$

$$B_I = \sqrt{\frac{E_I T_S}{N_0 T_I} \left(\frac{T_I}{T_S} - N_I \right)} ;$$

fering with the desired symbol.

= fractional part of a Q-channel chip, interfer-

$$B_Q = \sqrt{\frac{E_Q T_S}{N_0 T_Q} \left(\frac{T_Q}{T_S} - N_Q \right)} ;$$

ing with the desired symbol.

The total interference power, P_i/N_0 , is related to E_I/N_0 and E_Q/N_0 by where

$$\frac{E_I}{N_0} = \frac{P_i T_I}{N_0} \left(\frac{P_I}{P_i} \right)$$

$$\frac{E_Q}{N_0} = \frac{P_i T_Q}{N_0} \left(\frac{P_Q}{P_i} \right)$$

P_I = I-channel power of the interferer.

P_Q = Q-channel power of the interferer.

The characteristic function above assumes that

the interference is tuned to the same frequency as the desired signal. If the interfering signal is off-tuned from the desired signal, then filtering may significantly reduce the interfering signal power. The interfering power that is not filtered is assumed to be centered at the carrier frequency of the desired signal.

TDRSS signals are either uncoded, 1/2 or 1/3 rate convolutionally coded. Using the characteristic function, the P_e can be determined for each of these cases. In the uncoded case the P_e is given by

The inner integral is a Fourier transform that is

$$P_e = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} C(u) e^{j u z} du dz$$

used to change the characteristic function into a density function. The outer integral is used to determine the P_e from the density function.

In the coded case, the decoder is assumed to consist of an eight level quantizer and Viterbi decoder. The P_e is calculated from a heuristic expression [1] based on the computational cutoff rate R_0 . The parameter R_0 is given by where

$$R_0 = -\log_2 \sum_{i=1}^8 \left\{ \frac{1}{2} (P(i+1)^{1/2} + P(i-1)^{1/2}) \right\}^2$$

$P(i+1)$ = the transition probability that the transmitted bit is positive and the i th quantizer level is received. Since the channel is assumed to be symmetric, a negative transmitted bit has the same transition probabilities as the positive bit, but in reverse order.

The optimum step size between quantization

$$P(i+1) = P((9-i)-1) = \int_{\text{ith quantizer interval}} \int_{-\infty}^{\infty} C(u) e^{j u z} du dz$$

boundaries for an 8-level quantizer has been determined for a Gaussian channel to be $\sigma/2$, where σ represents the standard deviation of the noise component at the output of the coherent

integrate and dump detector.

The probability of error equation was developed heuristically, by adjusting constants so that the equation matched the results from a Gaussian noise channel simulation of a Viterbi decoder. The equation that results is shown below for both rate 1/2 and 1/3 constraint length 7 convolutional codes.

where

$$\ln(P_e) = \ln(K_1(R)) + \frac{K_2(R)}{R} \ln \left[\frac{2^{1-R_0} - 1}{1 + e^{a(R_0-b)}} \right]$$

R = code rate

a = 3.017

b = 1.602

K1(1/2) = 102.3 K1(1/3) = 42.284

K2(1/2) = 5.834 K2(1/3) = 5.918

Averaging over all phases of the interfering signal as described above is a reasonable approximation to the actual interference situation, where different Doppler rates result in a time varying phase of the interference signal relative to the desired signal.

B. Interference Threshold Angle Calculator

The Interference Threshold Angle Calculator computes threshold angles required to meet the desired discrimination. These angles are the inter-user angles (see Figure 2) between the desired user spacecraft and the interfering user spacecraft, as viewed from TDRS. It is assumed that the desired user spacecraft is directly in line with the TDRS antenna boresight. For Multiple Access (MA), this means that the beam is formed in the direction of the desired user spacecraft. Thus, the computed threshold angle is the minimum angular difference between the TDRS antenna boresight and the TDRS-to-desired look angle necessary to attain the required discrimination. However, it should be noted that since discrimination versus angle-off-boresight is not a strictly increasing function, the required discrimination may also be achieved for some angles less than the threshold angle.

The actual discrimination is the sum of gain discrimination and polarization rejection. The gain discrimination is acquired from the TDRS Antenna Pattern Databases. Although the actual TDRS MA antenna pattern and associated gain

discrimination will vary slightly with the position of the desired user spacecraft, the databased gain discriminations used by the ACRS model are accurate to within .2 dB for spacecraft altitudes under 1000 kilometers. Polarization rejection is only applicable if the signals from the desired and interfering user spacecraft are oppositely polarized.

Three types of threshold angles are computed, all of which are worst-case scenarios: data degradation, acquisition delay, and bit synchronization loss. These three types of angles are computed for all possible desired/interfering link combinations, and are computed for each TDRS using the method described above. An output value of zero for a threshold angle indicates that interference cannot possibly occur for the link combination. These calculated threshold angles are used as input to the Potential Interference Interval Calculator.

C. Potential Interference Interval Calculator

The Potential Interference Interval Calculator uses actual TDRS and user spacecraft orbital data to compute TDRS view periods for, and potential interference intervals between active TDRSS users. All TDRS view periods computed are based strictly on line-of-sight visibility. Data degradation and acquisition delay interference intervals are computed, corresponding to two of the three types of threshold angles computed by the Interference Threshold Angle Calculator. Interference intervals are computed for all selected TDRSS. For a given TDRS, an interference interval is defined as an interval of time for which the following conditions hold:

- 1) Both the desired and interfering user spacecraft have line-of-sight visibility to the given TDRS.
- 2) The inter-user angle between the desired user spacecraft and the interfering user spacecraft as viewed from the given TDRS is less than the appropriate threshold angle.
- 3) If the analysis is to consider the actual communications schedules for the user spacecraft, those schedules must exist in the ACRS schedule databases and include overlapping desired and interfering user service periods for the given TDRS.

Before each run of the Potential Interference Interval Calculator, the analyst must specify several run parameters. The most important of these are:

- 1) Starting Greenwich Mean Time (GMT) of the run.
- 2) Duration of the run.
- 3) Which TDRSs are active for the run.
- 4) Which user spacecraft are active for the run.
- 5) Whether or not scheduling data will be considered for the run.

The orbital data for each active TDRS and user spacecraft is obtained from the Orbital Information Database (OID). The orbital data is continually updated by received orbital information from the ISN. The state vectors from the OID are propagated forward to the next vector or the end Greenwich Mean Time (GMT) of the run. The orbit generator used is one whose force model only includes the oblate Earth. Assuming the spacecraft does not maneuver, the state vector error is less than 10 seconds per week of propagation.

Between the start GMT and the end GMT of the run, spacecraft state vectors and all of the geometric calculations are performed at discrete time points, ten seconds apart. Thus the start and stop times of view periods or interference intervals are only accurate to the nearest 10 seconds.

In addition to start times, stop times, and durations, the Potential Interference Interval Calculator provides other outputs for each interference interval. For each interval, possible mitigation techniques, if any, are provided. A mitigation technique is listed for an interval if it is both a possible option and will prevent the interference from occurring. However, there is no guarantee that employing a suggested mitigation technique will not cause an interference problem with another link. Suggested possible mitigation techniques include changing the frequency, polarization, or supporting TDRS for the desired or interfering user.

Additionally, for all data degradation interference intervals, a flag is provided indicating whether or not the interference event will probably be observable in real-time at White Sands Ground Terminal (WSGT) or the NCC. This

flag is set if the angle between the two user spacecraft is less than the appropriate bit synchronization loss threshold angle, predicting a loss of bit synchronization due to interference.

D. Solar Interference

ACRS predicts solar interference by treating the Sun in a manner similar to most other interfering user spacecraft. The Potential Interference Interval Calculator uses the desired user's position, the sun's position, and the appropriate Interference Threshold Angle to compute predicted intervals of solar interference which could cause data degradation, late acquisition, or loss of bit synchronization.

Computation of the interference threshold angles for solar interference differs from the standard computation as described in the section above on the Interference Threshold Angle Calculator. First, the required brightness temperature must be computed for each link of each desired user spacecraft. The required brightness temperature is defined as the brightness temperature which, when added to the normal system noise temperature, will reduce the desired user's link margin to zero. The link margins used are the same as those used in computing the required discrimination (see the section above on Calculation of Required Discrimination).

The required brightness temperature is then used to compute time-dependent solar interference threshold angles. This angle is defined as the minimum angular distance required between the TDRS receive antenna and the center of the sun to guarantee that the brightness temperature as seen by the TDRS antenna is less than the previously computed required brightness temperature. A different angle is computed for each month for each desired user spacecraft link. The time-dependence is necessary to allow for the different levels of solar activity at various times in the solar cycle. The solar interference threshold angle is computed by using a detailed model of the TDRS antenna pattern and a model of solar emissions in the appropriate frequency band during the desired month. The solar model is identical to that used in the CLASS Solar Interference Analysis package.

V. DESCRIPTION OF ACRS OUTPUT

The output produced for a given ACRS analysis run is a mutual interference summary chart which can be displayed on the screen or printed out on a local printer. A subset of one of these charts is shown in Figure 4. The summary charts are divided into 24-hour periods referenced as days. There are three tables displayed per TDRS for each day:

- TDRS view periods
- Acquisition delay interference intervals, and
- Data degradation interference intervals.

The view period table lists all of the user spacecraft considered and the date and time for

the beginning and end of each mutual interference interval. The format of the acquisition delay and data degradation tables are, for the most part, the same. The desired user is identified followed by columns of information for each possible mutual interference occurrence. The first three columns identify the start, stop, and duration of the interference interval, respectively. The data degradation table contains an additional column identifying, with a double asterisk, the predicts of interference severe enough to be visible to the NCC or WSGT during real-time operations. The next column identifies the operational link of the desired user as defined in appendices attached to each ACRS output. The interfering Link ID and

CLASS AUTOMATED CONFLICT RESOLUTION SYSTEM ANALYSIS #561				06/29/93			
TDE VIEW PERIODS FOR DAY 1							
4377	6/29/1993	0:0:0	6/29/1993	0:2:4	4		
3782	6/29/1993	0:0:0	6/29/1993	0:24:1	3		
6951	6/29/1993	0:0:0	6/29/1993	0:58:3	5		
3368	6/29/1993	0:6:3	6/29/1993	1:1:7	2		
4446	6/29/1993	0:24:5	6/29/1993	1:1:2	2		
6255	6/29/1993	0:31:5	6/29/1993	1:1:1	1		
2057	6/29/1993	0:40:1	6/29/1993	1:1:1	1		
4446	6/29/1993	0:40:1	6/29/1993	1:1:1	1		
4377	6/29/1993	0:30:1	6/29/1993	1:1:1	1		
782	6/29/1993	1:1:3	6/29/1993	2:2:2	2		
9951	6/29/1993	1:41:2	6/29/1993	2:2:2	2		
6951	6/29/1993	1:47:1	6/29/1993	3:3:3	3		
4446	6/29/1993	2:2:2	6/29/1993	3:3:3	3		
6255	6/29/1993	2:8:3	6/29/1993	3:3:3	3		
2057	6/29/1993	2:13:1	6/29/1993	3:3:3	3		
4446	6/29/1993	2:47:1	6/29/1993	3:3:3	3		
4377	6/29/1993	3:5:5	6/29/1993	4:0:0	0		
3782	6/29/1993	3:31:1	6/29/1993	4:0:0	0		
6951	6/29/1993	3:22:1	6/29/1993	4:0:0	0		
3368	6/29/1993	3:4:4	6/29/1993	4:0:0	0		
4446	6/29/1993	3:52:2	6/29/1993	4:0:0	0		

CLASS AUTOMATED CONFLICT RESOLUTION SYSTEM ANALYSIS #561				06/29/93			
ACQUISITION DELAY INTERFERENCE INTERVALS FOR TDE ON DAY 1							
START GMT:	6/29/1993	0:0:0					
FINAL GMT:	6/29/1993	4:0:0					

DESIRED USER: 1396							
START TIME	END TIME	DUR (SEC)	DESIRED LINKID	INTERFERING LINKID	USER	POSSIBLE MITIGATION	
000000	010700	20	MMLL002K002K	MMLH001K001K	446	NONE	
000000	010700	20	MMLL002K002K	MMLH004K004K	446	NONE	
000000	010700	20	MMLL002K002K	MMLH004K032K	446	NONE	
000000	010700	20	MMLL002K002K	SMLH001K001K	446	NONE	
000000	010700	20	MMLL002K002K	SMLH004K004K	446	NONE	
000000	010700	20	MMLL002K002K	SMLH004K032K	446	NONE	
000000	020000	20	MMLL002K002K	MMLH001K001K	446	NONE	
000000	020000	20	MMLL002K002K	MMLH004K004K	446	NONE	
000000	020000	20	MMLL002K002K	MMLH004K032K	446	NONE	
000000	020000	20	MMLL002K002K	SMLH001K001K	446	NONE	
000000	020000	20	MMLL002K002K	SMLH004K004K	446	NONE	
000000	020000	20	MMLL002K002K	SMLH004K032K	446	NONE	
000000	020000	20	MMLL002K002K	SMLH001K001K	446	NONE	
000000	020000	20	MMLL002K002K	SMLH004K004K	446	NONE	
000000	020000	20	MMLL002K002K	SMLH004K032K	446	NONE	
000000	020000	20	MMLL002K002K	MMLH001K001K	446	NONE	
000000	020000	20	MMLL002K002K	MMLH004K004K	446	NONE	
000000	020000	20	MMLL002K002K	MMLH004K032K	446	NONE	

Figure 4. Sample ACRS Output

user are displayed in the following two columns. The final column contains any possible options for interference mitigation. These options are described in detail in an ACRS output appendix. (For detailed information on ACRS output formats, see [3]).

VI. CONCLUSION

As the number of TDRS user spacecraft increases, so does the potential for interference arising from two or more spacecraft communicating simultaneously with the same TDRS. ACRS is a new tool used for mission-current mutual interference prediction, and although it is still in the operational prototype stage, the implementation of ACRS in the NCC OCR is already being used as an aid for communications scheduling. Future enhancements to the program, already under development, include forward link mutual interference prediction and refined mutual interference algorithms based on the results of ongoing validation studies of ACRS outputs vs. actual observed mutual interference events. ACRS will help the NCC accomplish the ever more challenging job of scheduling uninterrupted communications for NASA missions.

REFERENCES

- [1] A. Weinberg, "The Impact of Pulsed RFI on the Coded BER Performance of the Nonlinear Satellite Communication Channel," *IEEE Transactions on Communications*, Vol. Com-29, No. 5, May, 1981
- [2] R. R. Bate, D. D. Mueller, J. E. White, *Fundamentals of Astrodynamics*, New York: Dover Publications, 1971
- [3] "CLASS ACRS/TLAS Operator's Manual and Reference DRAFT," Stanford Telecom: 004R1693, June, 1993.
- [4] T. Kaplan, et al, "CLASS Interference Analysis System," *IEEE National Telesystems Conference*, May, 1992.
- [5] Yen F. Wong and James L. Rash, "An Interference Mitigation Methodology for Scheduling in Space Communications," *1990 Research and Technology*, NASA GSFC, 1990.



THE THREE-DIMENSIONAL EVENT-DRIVEN GRAPHICS ENVIRONMENT
(3D-EDGE)

By Jeffrey Freedman, Roger Hahn and David M. Schwartz

Stanford Telecom
7501 Forbes Blvd
Seabrook, MD 20706
(301) 464-8900

ABSTRACT— Stanford Telecom developed the Three-Dimensional Event-Driven Graphics Environment (3D-EDGE) for NASA Goddard Space Flight Center's (GSFC) Communications Link Analysis and Simulation System (CLASS). 3D-EDGE consists of a library of object-oriented subroutines which allows engineers with little or no computer graphics experience to programmatically manipulate, render, animate, and access complex three-dimensional objects.

Figure 1 shows the 3D-EDGE data hierarchy. The figure shows the 3D-EDGE data dependencies, and the supporting subroutines for each data type. The remainder of this section describes Figure 1 in detail.

I. INTRODUCTION

3D-EDGE was developed to allow programmers with little or no computer graphics experience to incorporate three dimensional solid objects into their programs. Other programmatic graphic interfaces [1,2] such as PHIGS require the user to have an in-depth knowledge of the type of object being modeled and how it is manipulated. This limits programmatic access of three-dimensional objects to people who have three dimensional computer graphics training. 3D-EDGE, on the other hand, uses very simple commands to manipulate, access and render the solid model. The user only needs to learn a few 3D-EDGE commands to use any three dimensional object because 3D-EDGE has the same generic interface for every object. This allows the user to access objects without knowledge of their internal data structure.

II. OVERVIEW

3D-EDGE can incorporate a wide variety of solid model representations, keeping their internal structures invisible to the user. Thus, once a user is familiar with 3D-EDGE, he/she can manipulate, render, animate and access any object regardless of its internal configuration.

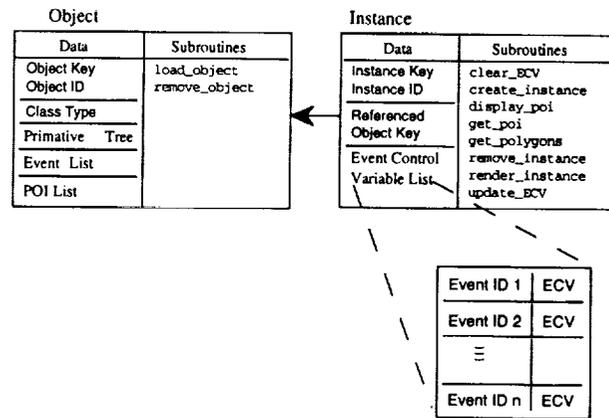


Fig. 1. 3D-EDGE Data Hierarchy Diagram

The database contains a description of a three dimensional object, an associated list of *events*, and a set of Points of Interest, (*PoI*'s, these are discussed below). *Events* control location, orientation, and any other configurable features of the model. For example, CLASS's model of the Space Shuttle contains *events* which control its location, orientation, the percentage Shuttle doors are opened, and the gimbal angles of the antennas. The Shuttle doors are opened by changing an appropriate *event's* value (e.g. "DOORS OPEN" *event*).

An *instance* contains a list of *Event Control Variables* (*ECV*'s) that specify values for every *event*. Since the user never modifies the object itself, only the *instance*, multiple *instances* of the same object can be controlled simultaneously

while maintaining data integrity and minimizing memory usage. The user updates the *ECV's* through a set of 3D-EDGE subroutines. The configuration of the instance is calculated only when the user either renders (graphically displays the *instance* of an object) or accesses information from the *instance*. Dynamic motion and animation can be effected by interactively changing *ECV's* and re-rendering.

The user accesses information about an *instance* of an object by calling an appropriate query routine. The query returns to the user information which may include, but is not limited to, a polygonal description of each surface on the object, a surface color, a surface dielectric constant, and a *PoI*. A *PoI* is a location and direction on an object which moves with the object. For example, a *PoI* can be the location and direction of an antenna boresight, which automatically moves with the antenna.

Figure 2 shows an overall flow diagram of a 3D-EDGE program. The user first loads an *object* from the database. Then one or more instances of the *object* are created. The instance is modified by using the update_ECV subroutine, which changes a particular ECV in an *instance*. For example, an update_ECV can be used to open Shuttle doors or to gimbal antennas. Next, either information about the *instance* (e.g. polygonal locations or dielectric constants) may

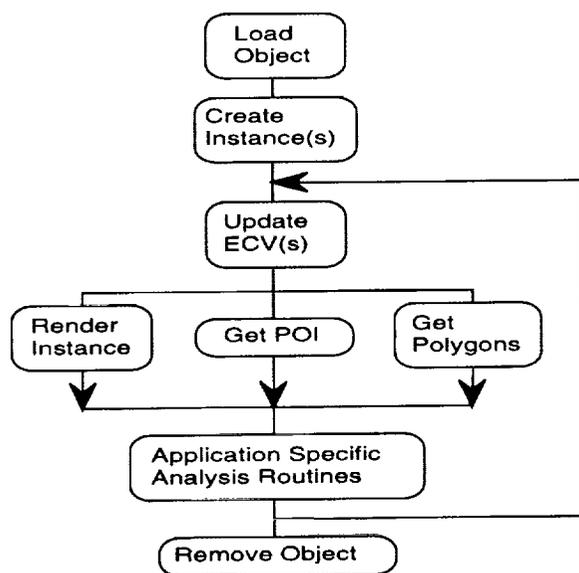


Fig. 2. 3D-EDGE Flow Diagram

be accessed or the *instance* rendered. This process can then be repeated or the *object* is disposed of if it is no longer required.

III. OBJECTS

An *object* refers to a three dimensional model. 3D-EDGE is designed to allow easy programmatic access to these *objects*. Each *object* belongs to a particular *class*. The *class* refers to the underlying solid model description of the *object*. For example, the *polygon class* (currently the only *class* supported) refers to objects made from polygonal surfaces. Additional proposed classes of objects include descriptions based on extrusion, B-spline and fractal solid model representations.

A. Hierarchy

Objects are comprised of a set of hierarchically related components known as *primitives*. *Primitives* are related to one another through transformation matrices. An example of this hierarchical structure is a simplified Space Shuttle *object* which is comprised of 8 *primitives*: a fuselage, a nose, a tail, two cargo bay doors, and a robotic arm comprised of three parts: a turret, forearm and clamp, as shown in Figure 3. The fuselage is the *root primitive* which has five children: the nose, tail, doors and turret. The shoulder in turn has one child, the forearm, and the forearm one child, the clamp.

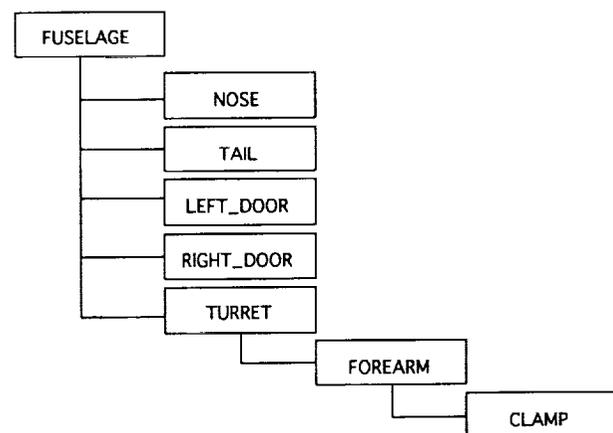


Fig. 3. Simple Shuttle Primitive Hierachy

The use of a hierarchical structure simplifies the model definition and provides a natural mechanism for specifying the manipulation of the model.

If an *event* is invoked to rotate the turret of the robotic arm, the rest of the arm moves as well. This is true for all parent-child pairs. As the parent moves, so do the children and the children's children, on down the line. Thus, by rotating an *object's root primitive* the entire *object* is effectively rotated in space.

B. Attributes

Attributes are used to define an *object's* physical characteristics. Examples of attributes include color, reflective coefficient, and dielectric constant. An *object* inherits a set of attributes from 3D-EDGE based upon the *object's class*. An *object's attributes* are defined within the *object's database*.

C. Inheritance

Inheritance, an important feature in 3D-EDGE, takes two forms: *events* and *attributes*. Both are inherited from more general levels to more specific levels. For example, if an *object's root primitive* is defined to be "white" in the *database*, then the rest of the *object* is considered to be "white." However, if a particular *primitive* has "blue" as the value of its *color attribute*, that *primitive* and any of its descendants will also be "blue" unless they too re-specify the *color attribute*.

D. Points of Interest

A *Point of Interest* ("PoI") is a location and direction on a *primitive* which moves with the *primitive*. By using 3D-EDGE's query language, the location and direction of the *PoI* may be retrieved at run-time. This allows programmatic use of the information. A program developed for CLASS uses a *PoI* to represent the antenna beam (boresight) of a satellite's gimbal antenna. As the antenna moves, so does its boresight. The *PoI* information is then used to control the camera (the user's view) in a real-time graphics package. The scene is then viewed from the antenna along its boresight.

IV. EVENTS

Each object has an associated list of *events*. *Events* are used to control and manipulate the *object*. They control location, orientation and

any other variable attributes of the model. *Events* are characterized by the following parameters: Type, Level, and Order.

This concept of *events* lies at the heart of 3D-EDGE because it means that the user needs to know nothing about how the *object* is defined in order to manipulate it. From the user's perspective, the type or level of the *event* is irrelevant. The *event* need only be defined for the *object* and invoked by the user; 3D-EDGE takes care of the rest.

A. Event Types

Events may be either simple or complex. For example, CLASS's model of the Space Shuttle contains *events* which control its attitude (orientation in space), the percentage the Shuttle doors are opened, and the gimbal (rotation) angles of the antennas. A simple event on the Shuttle would be a "ROLL". In real terms this means a rotation of the Shuttle about its positive x-axis. A complex *event* would be "OPEN_DOORS" which it entails rotating two different *object* components about their respective rotational axes.

B. Event Levels

Events are defined at three different levels: the *object* level, the *class* level and the *system* level. *Object* level events are *object*-specific and as such are defined in the *object's database*. *Class* level events are those that are defined within 3D-EDGE code but only for a specific *class* of objects. *System* level events are events which are defined in 3D-EDGE system code for all objects. The "ROLL" *event* noted above is a *system* level event as it is defined for all objects. However, the "OPEN_DOORS" event is an *object* level event defined only for the Shuttle.

C. Event Order

Computer graphics algorithms typically use translation and rotation matrices to manipulate objects. These matrices must be applied in a set order to achieve the desired effect. For example, a translation issued before a rotation will place an object at a different location than the same rotation issued before the translation.

In 3D-EDGE, the user can specify events in any order. However, the order in which the

events are actually invoked is defined by the 3D-EDGE system. An example of this is the way Shuttle antenna gimbal angles are specified. Two separate angles are necessary to define the position of a gimballed antenna: azimuth and elevation. Mathematically, the order in which the transformations are performed on the *object's* data points is relevant. Therefore, within the database for the Shuttle the *events* "SET_AZIMUTH" and "SET_ELEVATION" are defined such that they will be performed in the appropriate order at run-time. However, the user can update the ECV's in any order and achieve the same results.

D. Event Control Variables (ECV)

ECV's are used to control events. The ECV's are specified as either an explicit value or as a percentage of the event's range. How the ECV is to be specified is *event*-dependent. An example of an *event* requiring an explicit value is the "SET_AZIMUTH" *event* defined above. When invoking the *event*, the user could specify an ECV of 240, which would mean "rotate the antenna about the appropriate axis two-hundred forty degrees." The "OPEN_DOORS" *event* for the Shuttle is an example of an *event* whose ECV is to be specified as a percentage. When invoking the *event*, the user could specify an ECV of 50, which would mean "open the Shuttle bay doors halfway." The full range of motion of the doors is defined in the *object's* database.

V. INSTANCES

The user creates an *instance* of an *object* in order to control, access and render the *object* at run-time. The *instance* and its associated sub-routines are illustrated in Figure 1. An *instance* of an *object* consists of a pointer to the instance (called an *instance key*), an *instance* identifier, a pointer back to the *object*, (called an *object key*), and a set of ECV's for all *events* defined for the *object*. It is the *instance* that is used to either render an object or access information about the object.

Since the user never modifies the *object* itself, multiple *instances* of the same *object* can be controlled simultaneously while maintaining data integrity and minimizing memory usage.

VI. SUBROUTINES

Subroutine calls are the vehicle through which the 3D-EDGE system routines are accessed. There are subroutines for loading *objects*, invoking *events*, and querying for information about *objects*. Interfaces to the subroutines are available for both C and FORTRAN. One of the subroutines that allows the user to get information about the object is "get_polygons". The "get_polygons" subroutine returns to the program the transformed data points describing the current configuration of an *instance* of an object. By passing a mask which describes the information to be extracted, "get_polygons" can be used to access other information about the object like *dielectric constants* and *color*. Although these attributes usually remain constant for the *object* at any configuration, it is often useful to access this type of information.

VII. EXAMPLE PROGRAM

Figure 4 contains a sample program. The program first loads the Space Shuttle object by passing the name of the database containing a model of the Space Shuttle, "shuttle_file," to the "load_object" subroutine which then returns an integer *object key*, "object_key." "object_key" is then used by the "create_instance" subroutine to create two different *instances* of the Shuttle. "create_instance" is invoked by passing to it the "object_key," specifying an "instance_id" ("DISCOVERY" or "COLUMBIA" in this case), and specifying a load preference. An integer *instance key*, used to reference the instance in the remainder of the program, is then passed back.

Once the two *objects* are *instantiated* the program loops 100 times changing the variable *i* from 0 to 99. Within the loop, "DISCOVERY", identified by its *instance key*, "Discovery_key," and "COLUMBIA," identified by its *instance key*, "Columbia_key," have their configurations altered by invoking specific *events*. Specifically, "DISCOVERY" has its doors opened, is pitched, and is yawed by *i*, *i/5*, and *i* respectively. "COLUMBIA" only has its doors opened $i*2$ percent of their range of motion. (Note: Since the "OPEN_DOORS" event is defined with "HARD_LIMITS" of $0 < x < 100$, for any value

```

/* LOAD OBJECT */
object_key = load_object(shuttle_file);
/* INSTANTIATE SHUTTLE */
Discovery_key = create_instance(object_key,
    "DISCOVERY",LOAD_ABSOLUTE);
Columbia_key = create_instance(object_key,
    "COLUMBIA",LOAD_ABSOLUTE);
/* LOOP */
for (i=0;i<100;i++){
    update_ECV(Discovery_key,"OPEN_DOORS",i);
    update_ECV(Discovery_key,"PITCH",i/5);
    update_ECV(Discovery_key,"YAW",i);
    update_ECV(Columbia_key,"OPEN_DOORS",i*2);
    get_polygons(Discovery_key,VERTEX_NORMAL |
        DIELECT, polygon_points_array1,
        num_points1);
    get_polygons(Columbia_key,VERTEX_NORMAL |
        DIELECT, polygon_points_array2,
        num_points2);
    process_polygons(polygon_points_array1,
        num_points1);
    process_polygons(polygon_points_array2,
        num_points2);
}

```

Fig.4. Sample Program

of i or $i*2 > 100$, the doors will only be opened the maximum of 100 percent.) Once all of the ECV's have been changed, subroutines are called which will cause the *instances'* configurations to be calculated. In this case, the subroutine is "get_polygons." This subroutine causes the polygons, their vertex-normals, and their dielectric coefficients to be passed back to the program. The first call calculates the points specifying the polygons for "DISCOVERY" and passes back "num_points1" points in the "polygon_points_array1" array. Similarly, the next call to "get_polygons" returns the "num_points2" points defining "COLUMBIA" in the array "polygon_points_array2." The program then calls its own routine "process_polygons" to do the actual analysis desired.

VIII. RESULTS

Figure 5a shows a model of the Compton Gamma Ray Observatory (GRO) satellite which was incorporated into 3D-EDGE. 3D-EDGE can render GRO, move it, change its orientation, gimbal the antennas, and rotate the solar panels. The solar panels are rotated by first creating an *instance* of GRO, and then updating the "ROTATE_PANELS" EVC. Figure 5b is the

GRO satellite after the solar panels have been rotated.

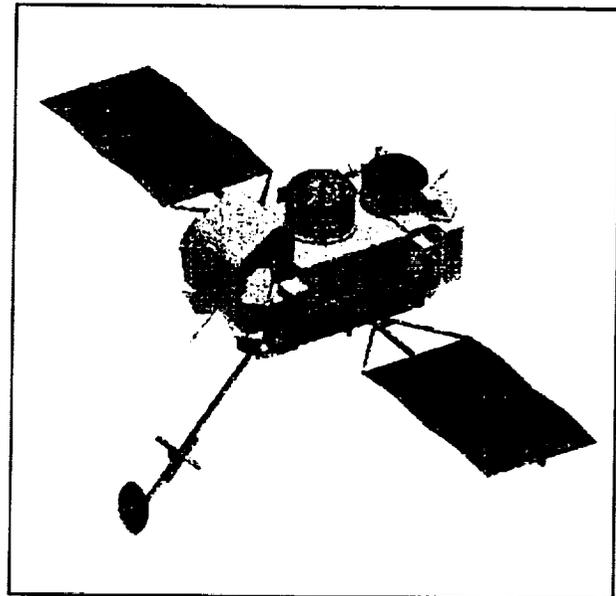


Fig. 5a. GRO before "ROTATE_PANELS"

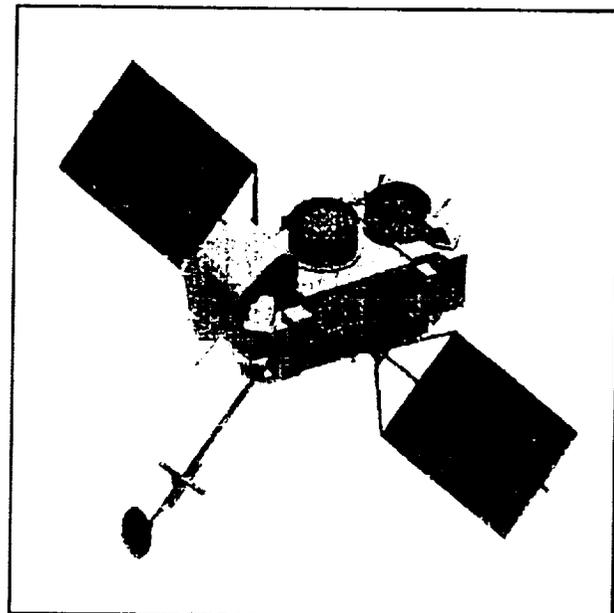


Fig.5b. GRO after "ROTATE_PANELS"
by +50 Degrees

CLASS uses 3D-EDGE in several of its programs to manipulate and render solid models. The CLASS Multi-Path Program (MPP) currently uses 3D-EDGE to control a three dimensional description of the spacecraft it is analyzing. The MPP has a minimal software interface

to 3D-EDGE because a single query of the model returns the necessary information (the dielectric constants and locations for each surface). The 3D-EDGE generic interface also allows the MPP to interchangeably use any spacecraft. Additionally, the CLASS Flight Performance System (FPS) uses 3D-EDGE for a graphical display of multiple models during a simulated Shuttle mission.

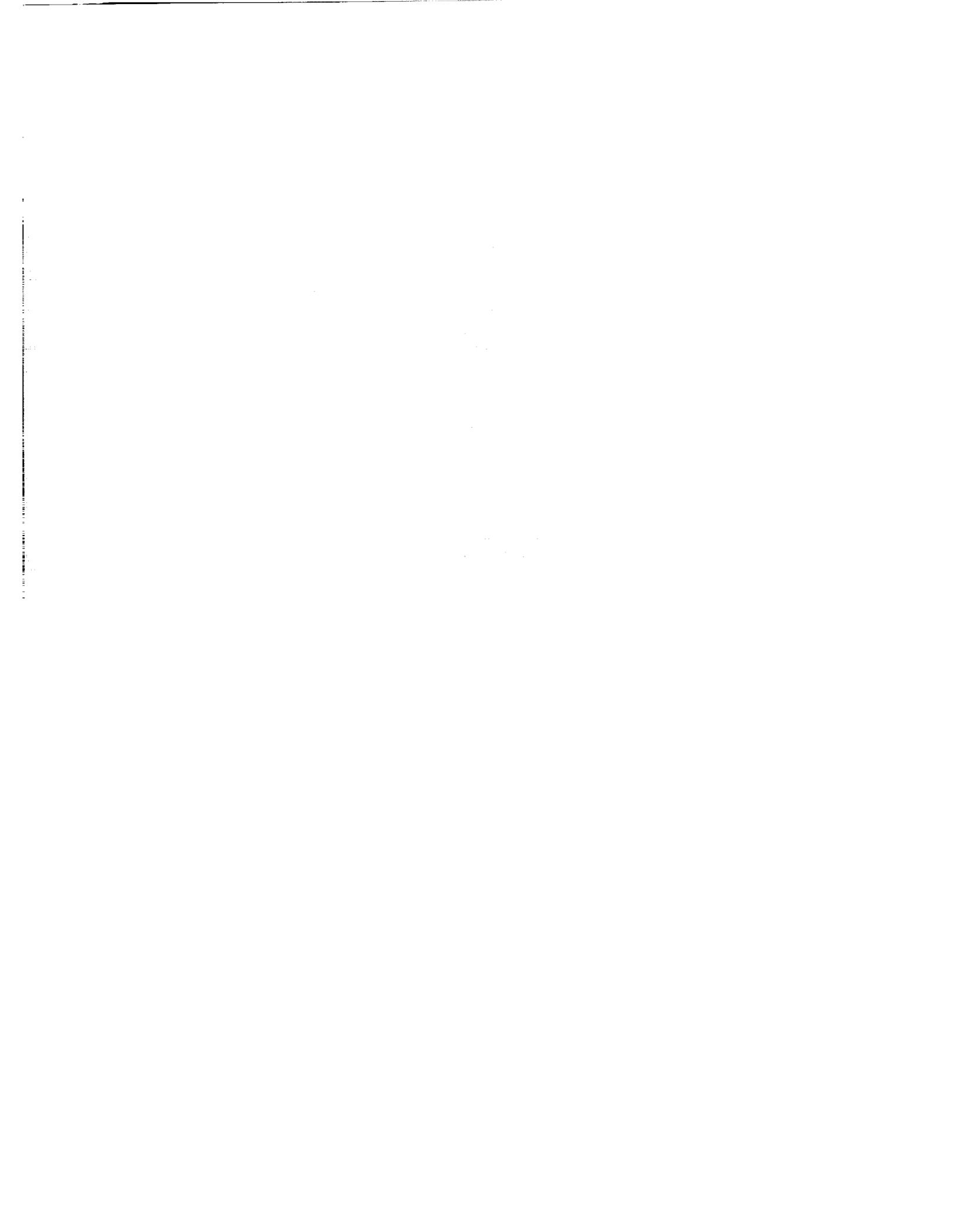
Before the development of 3D-EDGE, CLASS analysis programs that used solid models had severe limitations. These limitations included long development times, non-portable applications, lack of solid model data integrity between programs, and programs that could not interchangeably use different objects. 3D-EDGE solves these problems with an easy-to-use standardized graphics environment.

IX. CONCLUSION

3D-EDGE was designed on the principle that it is more important and efficient to spend time in the definition of a three-dimensional model than in the incorporation of that model into software. 3D-EDGE requires that when an object's database is being developed the events be defined along with the solid model description of the object. However, once defined, an object can be used by anyone with a knowledge of 3D-EDGE, even if they have minimal knowledge of 3D graphics. Further, by using abstract events and classes of data, virtually anything can be modeled and manipulated using only a small set of subroutines.

REFERENCES

- [1] T.L.J. Howard, W.T. Hewitt, R.J. Hubbard, K.M. Wyrwas, *A Practical Introduction to PHIGS and PHIGS PLUS*, Addison-Wesley Publishing Company, 1991.
- [2] *Starbase Graphics Techniques*, Hewlett Packard Company, 1991
- [3] N. Magnenat-Thalmann, and D. Thalmann, *New Trends in Animation and Visualization*, John Wiley and Sons, New York, 1991
- [4] J. Freedman, *Video Compression*, Ph.D. Dissertation, University of Maryland, May, 1991
- [5] J. Freedman, T. Kaplan, "Layered Command Driven Animation," Proceedings of SIGGRAPH 1992, July 1992 (In Review)
- [6] T. Kim, "Technical Reference for C.L.A.S.S. Vehicle Multipath Modeling Program," Stanford Telecommunications, Seabrook MD, May, 1991
- [7] C.S. Park, *Interactive Microcomputer Graphics*, Addison-Wesley Publishing Company, 1985
- [8] T. Budd, *An Introduction to Object-Oriented Programming*, Addison-Wesley Publishing Company, 1991
- [9] J. Freedman, R. Hahn, D. Schwartz, "The Three Dimensional Event Driven Graphics Environment (3D-EDGE)," IEEE National Telesystems Conference, May 1992.





**DEVELOPING A SPACE NETWORK INTERFACE SIMULATOR:
THE NTS APPROACH**

Gary E. Hendrzak
Booz • Allen & Hamilton Inc.
7404 Executive Place
Suite 500
Seabrook, Maryland 20706
Tel: (301) 805-5423
Fax: (301) 805-9535

ABSTRACT

This paper describes the approach used by Booz•Allen & Hamilton to redevelop the Network Control Center (NCC) Test System (NTS), a hardware and software facility designed to make testing of the NCC Data System (NCCDS) software efficient, effective, and as rigorous as possible prior to operational use. The NTS transmits and receives network message traffic in real-time. Data transfer rates and message content are strictly controlled and are identical to that of the operational systems. NTS minimizes the need for costly and time-consuming testing with the actual external entities (e.g., the Hubble Space Telescope (HST) Payload Operations Control Center (POCC) and the White Sands Ground Terminal). Discussed are activities associated with the development of the NTS, lessons learned throughout the project's lifecycle, and resulting productivity and quality increases.

INTRODUCTION

NASA's Spaceflight Tracking and Data Network (STDN) provides continuous telecommunications coverage for low-earth orbiting spacecraft such as the Space Shuttle, the HST, and the Gamma Ray Observatory. The NCC, located at the

Goddard Space Flight Center (GSFC), serves as the interface between the STDN and its customers, who primarily use the network to retrieve science and telemetry data from these spacecraft. The NCC consists of automated planning, scheduling, fault isolation, performance monitoring, communications, and display

systems (collectively called the NCCDS) that manage and control the network's resources.

Although the STDN offers a set of standard services to all science users, the addition of new users and new STDN elements requires modifications to the 438,000 lines of source code in the NCCDS. The most recent major change to the NCCDS was driven by the integration of a new Tracking and Data Relay Satellite System (TDRSS) earth terminal in White Sands, New Mexico, the Second TDRSS Ground Terminal (STGT).

The STGT integration requires intensive testing of new NCCDS software as well as tests of the changes to the interfaces between the NCCDS and each STDN user. The existing test system developed prior to 1983 was coded in assembly language and could not fulfill these test requirements. In addition, the user interface was cumbersome and supported only a single tester. The alternative was to test with the operational sites, which posed unacceptable risks to ongoing support of high profile user missions.

Recognizing this situation, GSFC management commissioned the development of a new NTS in late 1989.

Figure 1 portrays the role of the NTS in the context of the STDN. Testers use the NTS to simulate and test all external interfaces to the NCCDS. The NTS can also validate operational scenarios, provide an off-line test platform for new NCCDS software releases, and collect a wide variety of test data for analysis by developers and operations personnel.

The testing process consists of three phases shown in Figure 2. The first phase involves the development of test scripts, messages, and timing delays to simulate actual operational scenarios. For example, schedule requests from the HST POCC would be sent to the NCCDS, validated, and acknowledged. In the second phase, the test scripts are transmitted over actual NASA communication (Nascom) lines to the off-line NCCDS, while logging all message traffic. In phase three, the message traffic is analyzed to verify that test objectives were met.

FIGURE 1
NTS IN CONTEXT

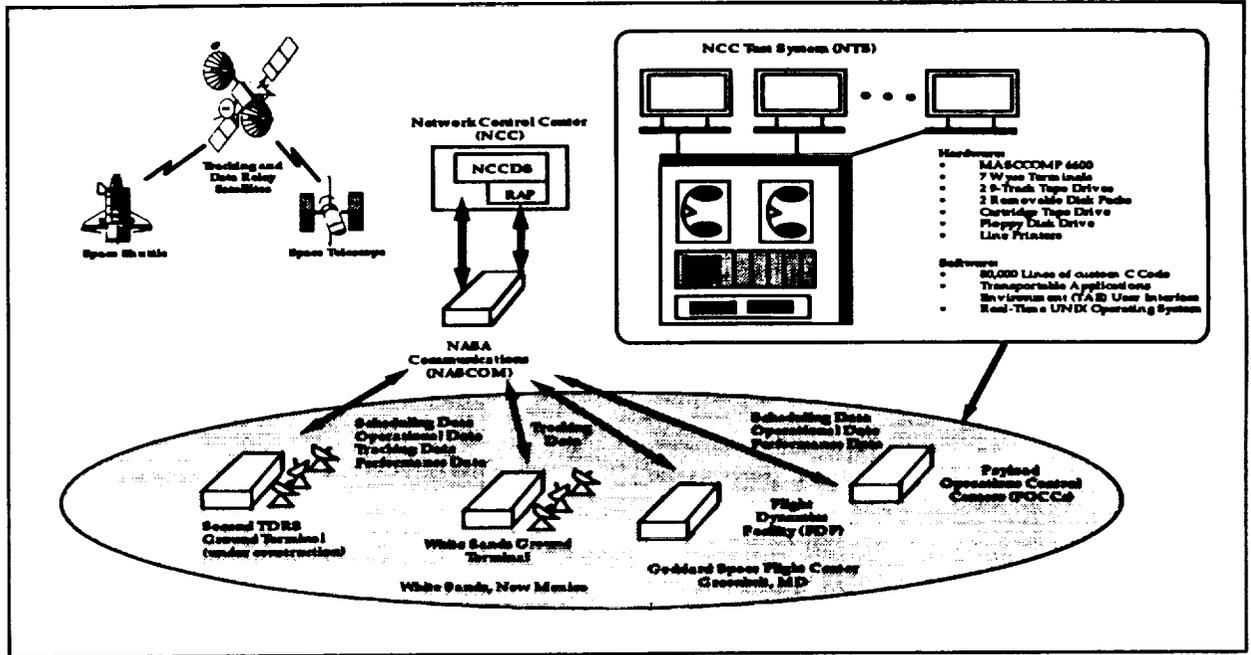
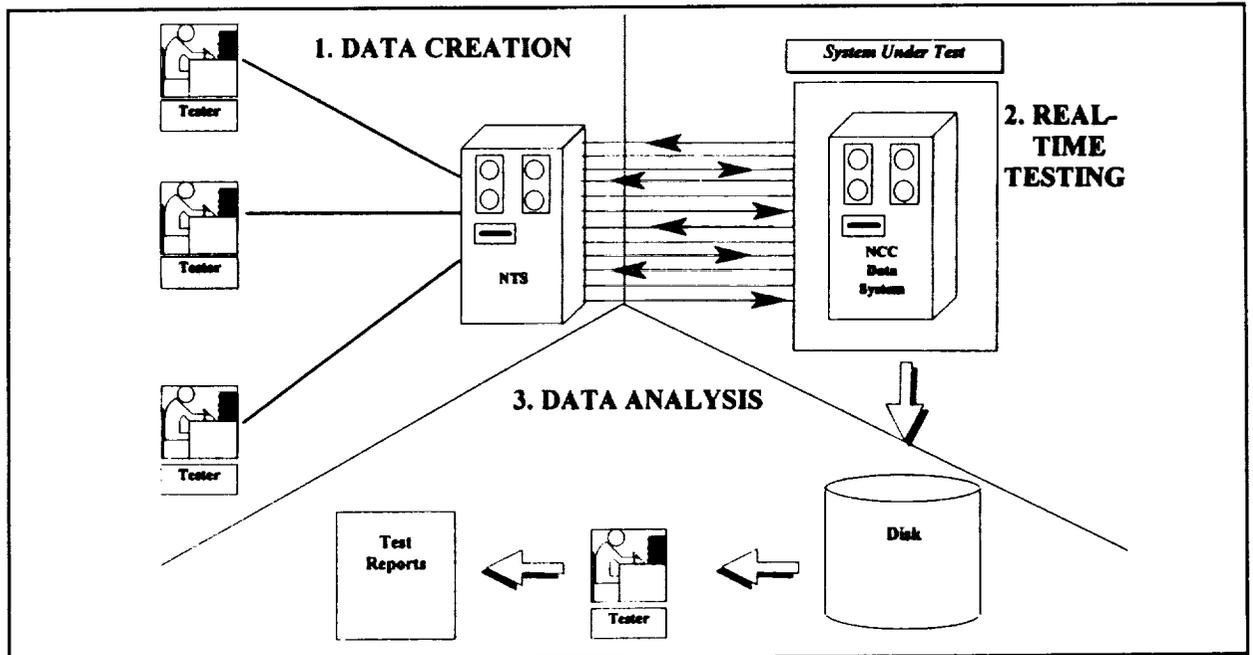


FIGURE 2
TESTING PHASES



DEVELOPMENT OBJECTIVES

In addition to those requirements documented in the System Requirements Specification [NTS93], several project goals were set. The system had to meet the schedule for the implementation of STGT-related changes in the NCCDS software. In addition, existing functionality and command syntax had to be replicated to minimize the learning curve required for NCCDS test teams. The human-machine interface had to be user-friendly and permit concurrent use. Finally, the NCCDS testing process had to be made more efficient by automating as many functions as possible. Likewise, the NTS development approach had to meet a set of objectives that the development team believed were vital:

- To perform only those tasks that directly added value and directly contributed to the success of the project
- To have end-users play a vital role throughout the project life-cycle
- To develop a system that promotes encapsulation, maintainability, modularity, extensibility, and re-use
- To define a process that is

measurable, manageable, and repeatable.

To meet these goals and objectives, a phased implementation schedule was selected. The first release of the system met these major goals: duplication of the present functionality, implementation of the new functionality required for STGT-related modifications, as well as a new human-machine interface. Subsequent releases were used to continually automate the testing process, save time, and support more rigorous scenarios.

DEVELOPMENT APPROACH

The NTS development approach followed the traditional waterfall model in which the process cascades from one level to the next in a smooth progression [SEL92]. While this is neither unique nor groundbreaking, the model was deemed sufficient to meet the project's goals. However, the development team realized that a number of inefficiencies embedded in the development process had to be justified or eliminated to stay on schedule. Examples include excessive amounts of documentation, inefficient configuration

management procedures, extended review and approval cycles, and responding to issues not relevant to the project. The development team believed that a more streamlined and flexible approach was preferable to the rigid, structured approach prescribed by the waterfall model. The philosophy of "Lean Software Development" described by Basili [BAS92] and based upon the work of Womack, et. al. [WOM90], seemed a perfect fit. This concept involves tailoring the development process to the needs of the product. Additionally, the Plan-Do-Check-Act cycle of Continuous Process Improvement espoused by Deming [DEM86] was applied to the development process, rather than to the product. As the project progressed, the entire process was continually refined and lessons learned were incorporated into subsequent development cycles.

Another key element in the approach was to include the NTS users in weekly functionality discussions and demonstrations aimed at specifying and clarifying new NTS requirements. The results of these meetings were captured and documented. Through numerous

discussions, the NTS development team gained an in-depth understanding of the users' needs. This knowledge and first-hand experience allowed both developers and users to recommend and refine a number of enhancements that saved time during test sessions, increased the quality of testing, and decreased the amount of human-intensive analysis that was common to the testing process. Not only was testing more efficient in the NCC, but the new NTS eliminated most of the preliminary testing sessions with each of the 34 external entities.

The development team also determined that the content of the design reviews was not directly adding value to the project. All too often, no substantive issues were raised at the reviews, mainly because the attendees were users concerned with *what* the system would do, and not *how* it was to be implemented. With the approval of GSFC management, the number and content of the reviews were tailored to explain the system from a user's perspective. System features were discussed, followed by a brief overview of their implementation. Finally, an operations concept of the feature was

presented using ToolBook™, a PC-based animation tool. These reviews, coupled with frequent human-machine interface demonstrations and a full day of hands-on training produced a system that exactly matched user expectations.

The development team also selected the Transportable Applications Environment (TAE) Classic for the user interface. TAE, developed for GSFC and maintained by Century Computing, consists of an interface that interacts with the user and manages the execution of application programs, while shielding the user from the host operating system. TAE provides a hierarchical menu system, on-line, context-sensitive help, parameter range checking, and a tutor mode to help new users build valid command strings. Thus, the users were required to learn only the NTS interface and not concern themselves with the operating system. By using TAE, the development team saved an estimated 630 staff days (approximately \$250K) of development effort. The single user problem was alleviated by hosting the system on a Masscomp 6600 computer. The Masscomp is a Unix-based timesharing system that supports 16

concurrent users.

The software, developed in C, was designed with reuse in mind. Various standalone programs were developed to assist the user in developing test data, changing the system configuration, and analyzing test results. The human-machine interface to all of these tools is common and contains over 7000 lines of reused software. In addition, a library of common functions was developed, containing over 3500 lines of code. In total, over 18% of the software was reused in subsequent releases to implement new functionality.

Due to the development team's close working relationship with the user group, problems were usually resolved and tested on the development system *prior* to receipt of the official documentation describing the problem. In addition, the team foresaw a problem associated with dual mode use (classified vs. unclassified). Sanitization of over 1.3 gigabytes of disk storage would require 4 hours. The team recommended removable disk drives, resulting in sanitization time being reduced to only 5 minutes.

RESULTS

The results of the principles applied on this project can best be described quantitatively. Figure 3 presents software error rates for the three development cycles. Table 1 presents software productivity metrics, based upon the philosophy of Putnam and Myers [PUT92]. These statistics suggest that continual refinement of the development process

FIGURE 3
SOFTWARE ERROR RATES

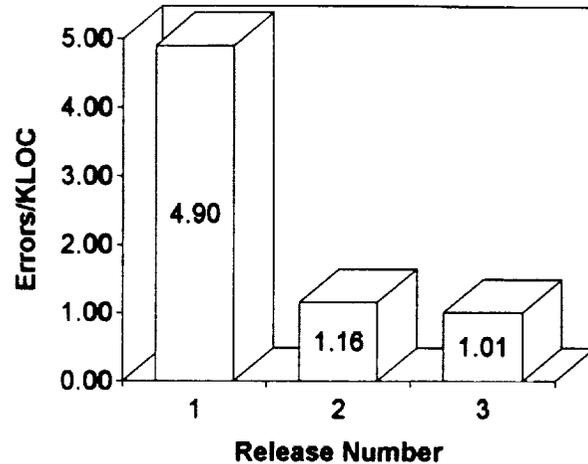


TABLE 1
SOFTWARE DEVELOPMENT METRICS

Version	Lines Of Code ¹ (LOC)	Staff Months ² (Effort)	Calendar Months ² (Time)	Productivity Parameter ³ (PP)	Productivity Index ⁴ (PI)
Release 1	22,023	83.0	14.0	5315	9
Release 2	26,773	89.5	11.0	8690	11
Release 3	21,836	53.0	9.5	10261	12

NOTES

1. Booz•Allen-developed software only.
2. Effort and Time are calculated from the beginning of the design phase (following the Software Requirements Review) until conclusion of the code/unit test phase.
3. The Productivity Parameter (PP) is calculated according to the following equation:

$$PP = (LOC)/(Effort/B)^{(1/3)} (Time)^{(4/3)}$$

4. B, the special skills factor, is a function of size. For all releases, B = 0.18.
The Productivity Index (PI) is obtained from Table 2.3 of [PUT92].

resulted in higher productivity and lower error rates. User satisfaction ratings of the tool's functionality, completeness of the User's Manual, and the quality of training continues to be high. Suggestions for system enhancements and additional functionality were prioritized and addressed in each new release with no impact to cost or schedule.

In May of 1993, the NTS development effort was selected as one of five representative software projects to be part of a Booz, Allen, corporate-wide software process assessment based upon the criteria developed by the Software Engineering Institute (SEI). The Institute has developed an instrument to assess an organization's software development process. The results of this self-assessment showed that the NTS development team was functioning as a Level 2 organization while exhibiting many of the qualities characteristic of a Level 3 organization. These results are significant because the SEI process assessment procedure is geared more toward larger, more functionally segmented organizations (e.g., those having separate configuration manage-

ment, quality assurance, test, and document preparation teams), whereas the NTS development team never consisted of more than 10 members.

CONCLUSIONS

Applying the principles of Lean Software Development and Continuous Process Improvement resulted in an increase in productivity and quality. This increase allowed for the delivery of additional functionality at no additional cost. Even though each release was successful, the development team continued to look for ways to improve and streamline the development process. Getting the user community involved from the very beginning and soliciting their input throughout the entire development process is a key strategy for success. Software development is by nature a dynamic process, constantly evolving and maturing. Change is a part of that process, and is not only necessary, it should be required.

The approach described here has resulted in the delivery of three separate releases of NTS software totaling 80,000 lines of source code. Each of these releases was

delivered on or ahead of schedule and 3-5% under budget. The NTS is functioning as intended, allowing testers to perform more robust and exacting tests on the target software. In fact, during the first few weeks of operational use, testers using the new NTS uncovered several defects in the NCCDS software that had not been discovered by its developers or during any of the previous independent test phases.

The NTS provides a significant increase in tester productivity over the previous system, permitting simultaneous test data creation, test execution, and results analysis. The system was designed and documented to support future growth and changing requirements. It is a user-friendly test tool, decreasing the overall certification time of the NCCDS software, while greatly improving testing accuracy.

ACKNOWLEDGMENTS

The author wishes to thank Clint Provenza, Hellmut Scheel, and Bill Brooks for their comments on earlier drafts of this paper. In addition, the input and guidance of Keiji Tasaki and Roger Clason, GSFC project managers, and of

course, the NTS user community, greatly contributed to the success of this project.

REFERENCES

- [BAS92] Basili, V., The Experience Factory: Can it make you a 5?... Proceedings of the 17th Annual Software Engineering Workshop, Goddard Space Flight Center, Greenbelt, MD, December, 1992.
- [DEM86] Deming, W., Out of the Crisis, MIT Center for Advanced Engineering Study, MIT Press, Cambridge, MA, 1986.
- [NTS93] NASA/GSFC, NTS Requirements Document, 530-SRD-NTS, February, 1993.
- [PUT92] Putnam, L. and W. Myers, Measures for Excellence, Yourdon Press, Englewood Cliffs, NJ, 1992.
- [SEL92] NASA/GSFC Software Engineering Laboratory, Recommended Approach to Software Development, SEL-81-305, August, 1986.
- [WOM90] Womack, J., et. al., The machine that changed the world: based on the Massachusetts Institute of Technology 5-million dollar 5-year study on the future of the automobile, Rawson Associates; New York, NY, 1990.



SATELLITE COMMUNICATION HARDWARE EMULATION SYSTEM (SCHES)

By Ted Kaplan

Stanford Telecom
7501 Forbes Blvd
Seabrook, MD 20706
(301) 464-8900

ABSTRACT — Satellite Communication Hardware Emulator System (SCHES) is a powerful simulator that emulates the hardware used in TDRSS links. SCHES is a true bit-by-bit simulator that models communications hardware accurately enough to be used as a verification mechanism for actual hardware tests on user spacecraft. As a credit to its modular design, SCHES is easily configurable to model any user satellite communication link, though some development may be required to tailor existing software to user specific hardware.

and extensions of the system have expanded the CLASS system capability to provide a general-purpose communications system analysis and design tool for use by both the network and the network user. CLASS models all elements of the network system, user system, and communications channel environment. It is capable of providing a rapid, reliable, and accurate performance analysis of virtually all communications system performance measures.

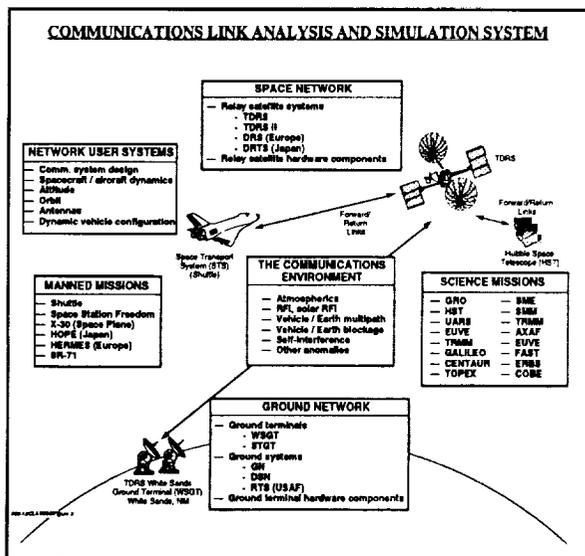
I. INTRODUCTION

The Communications Link Analysis and Simulation System (CLASS) has been developed by Goddard's Networks Division to provide a tool for evaluating the performance of space communication links through the network communications and tracking support elements, especially TDRSS. Subsequent enhancements

II. SCHES OVERVIEW

Most recently, the CLASS team has developed the Satellite Communication Hardware Emulator System (SCHES), a powerful simulator that emulates the hardware used in TDRSS links. SCHES is a true, bit-by-bit simulator that models communications hardware accurately enough to be used as a verification mechanism for actual hardware tests on user spacecraft. As a credit to its modular design, SCHES easily is configurable to model any user satellite communications link, though some development may be required to tailor existing software to user-specific hardware.

Hardware modules in the communication link are simulated effectively in SCHES using separate software modules. Each of these modules uses compatible input and output files which consist of data streams for the bit-by-bit simulation. The input file for any one hardware simulation module acts as the driver for that module. That module, in turn, produces an output file which drives the next module, while additionally allowing for the calculation of statistics at crucial points between modules. These analytical statis-



CLASS, developed to evaluate the performance of space communication links through network communications and tracking support elements.

tics provide otherwise unobtainable information on the performance of each individually modeled hardware subsystem. Finally, the individual simulation outputs are combined and analyzed to produce a complete and accurate representation of the proposed user satellite link.

This simulation approach requires the processing of statistically significant sample spaces which usually means much larger data bases than are required by an analytical approach. Nonetheless, there are powerful advantages to this true simulation approach: it serves not only as an analysis tool but also as a design tool, for the flexibility to alter individual channel elements enables to observe the effects these changes have on the overall channel performance. In particular, it affords us the ability to characterize the transient features of TDRSS.

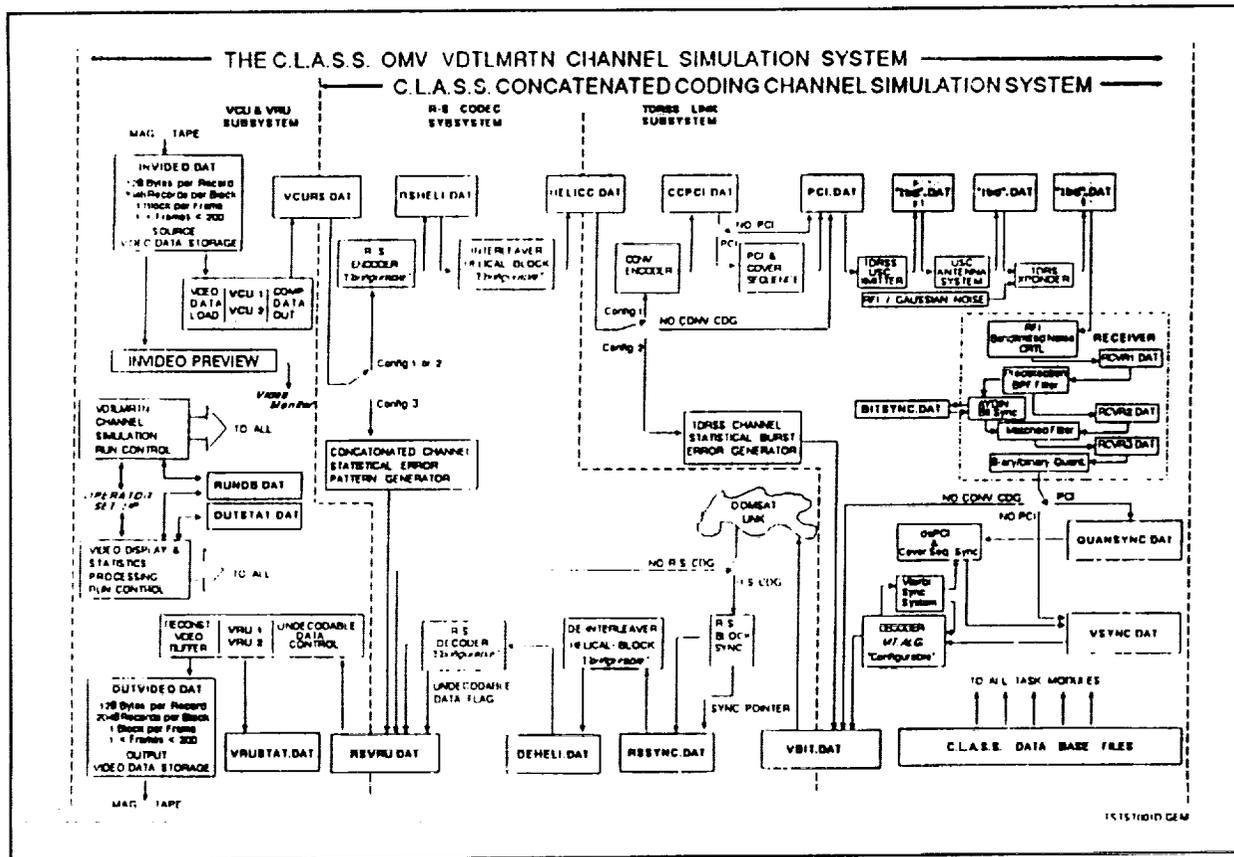
When large amounts of data have been collected on the behavior of a particular hardware module, a true hardware simulation for that hard-

ware subsystem may no longer be necessary. Instead, the simulation can be replaced with a functional model that uses appropriate statistics to corrupt the digital data stream. This functional model can provide the same accuracy as the direct emulation model, when predicting steady-state channel performance, but with the potential for enormously increased simulation run speeds.

The computational support for SCHES is provided by software hosted on an HP9000 computer, running under a UNIX operating system environment. The system includes a user-friendly interface for run control, provided on a Macintosh II. The capability to visually monitor test run activities is supported through the use of a video monitor.

III. IMPLEMENTATION FOR OMV

SCHES was tested during the course of a task to develop a complete model simulation of the Orbital Maneuvering Vehicle (OMV) video te-



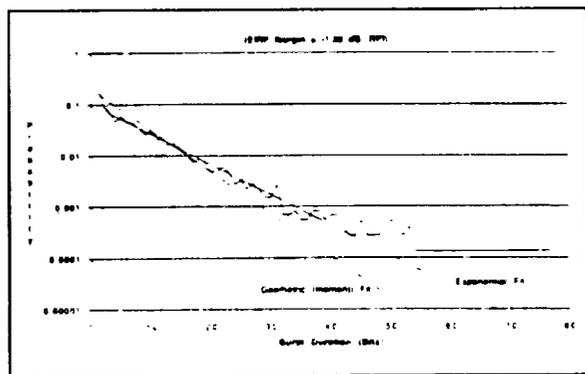
A block diagram of the CLASS channel simulation system.

lemetry return link. OMV was to be a remotely piloted spacecraft, designed to be part of the space transfer system.

The OMV video signal needed to be extremely robust to allow the pilot on the ground to view a target. Video compression and forward error correction, as described below, ensured the quality of the picture at the ground terminal. The camera's video signal was first digitized and compressed by 2-D differential pulse code modulation and Huffman coding. Error resistance was added through the use of Reed-Solomon encoding and Helical interleaving. A rate - 1/2 convolutional code was added with periodic convolutional interleaving so that the data could be relayed via TDRSS. Then, from White Sands Ground Terminal, the data was sent to Johnson Space Center via DOMSAT.

The pilot's commands to the OMV vehicle were transmitted by the forward link. Errors in the data transmission, however, were expected to result primarily from thermal noise and radio frequency interference (RFI) corruption of the TDRSS S-band return link between the OMV flight vehicle and the TDRSS spacecraft.

The essential concepts of the SCHES model of the OMV channel simulation are illustrated in the second figure.



Exponential and geometric fits to burst duration statistics

The model is separated into three subsystems: the video compression unit and video reconstruction unit, which are modules unique to OMV; the

Reed-Solomon coder-encoder subsystem; and the TDRSS link subsystem, which is part of standard CLASS. Each subsystem is further divided into modules. Each module simulates a hardware function and produces a data file which, in turn, drives the next module.

The DOMSAT link was not discretely modeled in the SCHES simulation because the BER through this link was reduced, through forward error correcting, to very low value. The other blocks in the system were exact, bit-by-bit hardware emulations of the actual system and together were used to characterize both transient (synchronization) behavior as well as static behavior of the channel.

IV. RESULTS

More than 20 simulations of the OMV return video link were completed, each requiring 25 hours of run-time. Runs were made with 50 frames apiece of data (approximately 5 million bits), and had varying effective isotropic radiated power (EIRP) margins and RFI environment conditions. The hardware simulation and the many test points provided the user with equivalent information to that acquired from actual hardware tests. Statistical processing was done by manipulating the data files after the simulation was over and by producing plots, histograms, and tables.

Statistics from different runs were plotted versus EIRP margin for each RFI condition. This data provided an easily understood statistical display of the actual performance characteristics of the video channel under varying environmental conditions.

Examples of some of the statistics produced are shown in the table and the third figure. These statistics are for an OMV communications link through TDRS-East, in a high RFI environment and with an EIRP margin of -1.5 dB. The fourth figure shows both the original picture frame (upper left), the reconstructed video display (lower right), as well as relevant channel statistics, as they appeared at run-time on the video monitor.

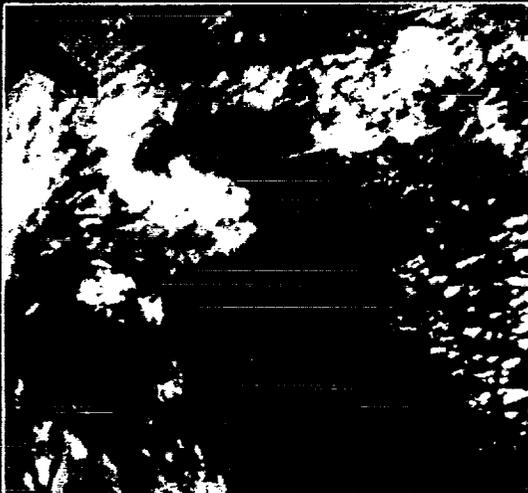
**Summary of OMV when Operating with a 1.5-dB EIRP Margin
in a Worst-Case TDRS-East Environment**

Channel Characteristics	Units	Value
Analysis ID	-	A908041411
RFI	-	SSA.TDRS.EAST
EIRP Margin	dB	-1.5
Data rate	Kbps	972
Number of lines per subframe	-	20
Initial stepsize	-	16
Number of frames	frames	50
Number of codewords transmitted	codewords	2390
Number of (convolutionally coded) symbols	symbols	9,751,200
Statistics Before DE-PEI	Units	Value
Mean Clock Jitter	% of symbol	-0.57
Standard Deviation of the Clock Jitter	% of symbol	2.18
Symbol Slip Rate	-	0
Random Error Rate	-	1.17E-2
Number of Bursts	bursts	186.577
Burst Window	symbols	12
Mean Burst Error Duration	symbols	13.68
Standard Deviation of Burst Error Duration	symbols	11.10
Mean Errors Per Burst	symbols	3.96
Standard Deviation of Errors Per Burst	symbols	2.43
Mean Space Between Errors in a Burst	correct symbols	3.54
Standard Deviation of Space Between Errors in a Burst	correct symbols	3.12
Error Rate Due to Burst	-	7.58E
<u>(# of Bursts) (Mean # of Errors Per Burst)</u> Number or Symbols Transmitted		
Total Error Rate = (Error Rate Due to Bursts) + (Random Error Rate)	-	8.7E-2
Transition Probabilities		
Pr(0/1)	-	.61274
Pr(1/1)	-	.12864
Pr(2/1)	-	.09934
Pr(3/1)	-	.06842
Pr(4/1)	-	.04940
Pr(5/1)	-	.02341
Pr(6/1)	-	.01091
Pr(7/1)	-	.00714
Pr(7/0)	-	.61286
Pr(6/0)	-	.12810
Pr(5/0)	-	.09973
Pr(4/0)	-	.07571
Pr(3/0)	-	.04215
Pr(2/0)	-	.02348
Pr(1/0)	-	.01075
Pr(0/0)	-	.00722
Predicted Viterbi Decoder Error Rate	-	2.64E-3
Analysis ID	-	A908041411

**Summary of OMV when Operating with a 1.5-dB EIRP Margin
in a Worst-Case TDRS-East Environment**

Statistics After DE-PEI	Units	Value
Random Error Rate		1.17E-2
Number of Bursts	bursts	198.021
Burst Window	symbols	12
Mean Burst Error Duration	symbols	14.93
Standard Deviation of Burst Error Duration	symbols	12.20
Mean Errors Per Burst	symbols	3.73
Deviation of Errors Per Burst	symbols	2.17
Mean Space Between Errors in a Burst	correct symbols	4.10
Standard Deviation of Space Between Errors in a Burst	correct symbols	3.08
Error Rate Due to Bursts	-	7.57E-2
Statistics After the Biterbl Decoded		
	Units	Value
Total Error Rate	-	5.219E-3
In-Sync Error Rate	-	5.219E-3
Number of Bursts	bursts	4730
Burst Window	bits	6
Mean Burst Error Duration	bits	7.96
Standard Deviation of Burst Error Duration	bits	6.84
Mean Errors Per Burst	bits	4.77
Standard Deviation of Errors Per Burst	bits	3.75
Longest Burst	bits	57
Statistics After the Reed Solomon Decoding		
	Units	Value
Undecodable Codewords In-Sync	Codewords	162
Undecodable Codewords Out-of-Sync	Codewords	28
Decodable Codewords In-Sync	Codewords	2209
Decodable Codewords Out-of-Sync	Codewords	0.0
In-Sync Codeword Error Rate	-	.068
First In-Sync Codeword	-	20
a) The first 8 codewords are dummy data used in initialize the helical interleaver. b) A codeword is declared in-sync when its sync counter value stays at 15 for two codewords.		
First Decodable Codeword After Declaring In-Sync	-	29
Number of Freewheeling Events	-	47
Freewheeling Value	-	14
Max Sync Counter Value	-	2
Number of Subframe Replacements During Initial Sync	Subframes	24
Number of Subframe Replacements After Initial Sync	Subframes	148
Total Number of Subframes	Subframes	600

C.L.A.S.S.: OMV VDTLMRTN ANALYSIS



CHANNEL MONITOR

Cal'n Sync- Psy.s.:	Cntr:
Code Sync- Pch.s.:	Cntr:
Data Sync- Psy.s.:	Cntr:
Uncoded EEP- Pe. (comp):	Cntr:
Vit Dec Sync- Out Bits:	Cntr:
Av Cov Seg Sync Search:	Cntr:
R-S Dec Sync- Max C.w.:	Shift:
		Full:
R-S Undec Data:	with Sync -	Cntr:
	without Sync -	Cntr:
Cur Fr Pch Err Std Dev:	
VRU Subfr Rpl L-S: S'fr:	
	Tel/Sim:	

SIMULATION PARAMETERS

Configuration: 2 3

33 Frames Considered of 100 Available

Current Video Frame: 1

TDRS: L W

EIRP: 24.47 dBW EIRP Margin: 1.0 dB

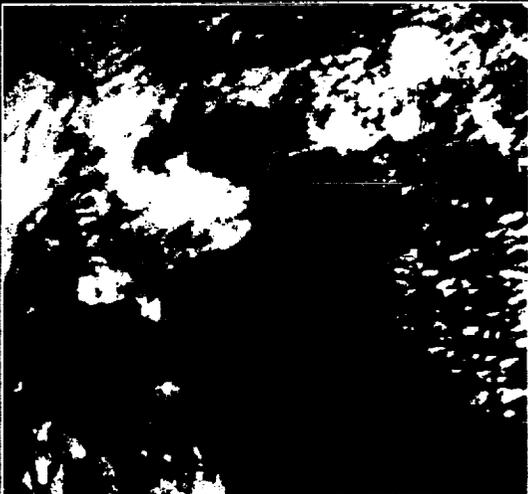
Range: 1000 km

RFI Cond: N 2 3 4 5 6 7

VCRS Model: FBI GSFC

VCU Mode: H B C Da Do

Channel ID: 1 2



The video monitor display for the OMV analysis.

**ORIGINAL PAGE IS
OF POOR QUALITY**

INTERFERENCE MONITOR

By Jeffrey Freedman and Catherine Ruseau

Stanford Telecom
7501 Forbes Blvd.
Seabrook, MD 20706
(301) 464-8900

ABSTRACT – Stanford Telecom developed the Interference Monitor (IM) for NASA Goddard Space Flight Center's (GSFC) Communications Link Analysis and Simulation System (CLASS). IM is a software program used to predict long term (i.e. 30+ years) statistics for mutual interference intervals of TDRS user spacecraft.

I. INTRODUCTION

TDRS user spacecraft periodically lose communication signals due to mutual interference. Mutual interference is defined as the interference between two spacecraft attempting to communicate with the same TDRS satellite at the same time. If the TDRS antenna discrimination is sufficient, two spacecraft can communicate at the same time with the same TDRS without mutual interference. However, when the user spacecraft appear close to each other (from point of view of TDRS), mutual interference may occur and communications can be lost.

The Interference Monitor (IM) was developed to predict long term statistics for intervals of mutual interference. IM simultaneously simulates the orbits of multiple user spacecraft while gathering interference statistics over long periods of time. IM can simultaneously simulate 100 user spacecraft orbits at 1 second intervals over a 30-year period. By examining many years of calculated orbits, IM can present an accurate statistical depiction of when, where and how much mutual interference will impair a user spacecraft's ability to communicate. The output plots and charts produced by IM provide NASA with accurate data for network and mission planning, interoperability studies and TDRS load analyses. What follows is an in-depth description of the analysis and the capabilities of IM.

II. ANALYSIS

IM uses an analytic pre-processing module and a simulation module to determine mutual interference statistics. The pre-processing module performs all the communications analysis in advance, and determines the conditions under which mutual interference can occur. The simulation module records statistics for user spacecraft as they meet these conditions.

The angle between two user spacecraft as seen from TDRS will determine if there is potential mutual interference between the two user spacecraft. This angle is called the inter-user angle and is shown in Figure 1. Separate antennas on TDRS communicate with each user spacecraft. The boresight of the TDRS antennas are pointed at the appropriate user spacecraft. As long as the inter-user angle is large, the interfering signals are transmitting to back lobes of the other antennas and mutual interference is negligible. However, when the inter-user angle is small, the interfering signals are transmitting to the main-lobe of the other antenna and communications can be lost.

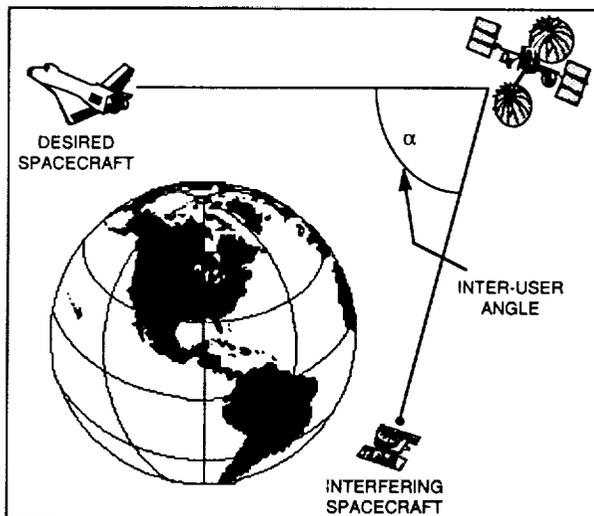


Figure 1. INTER-USER ANGLE - The angle, α , between the desired and interfering user spacecraft

Figure 2 is a block diagram describing the operation of the program. IM first determines the minimum inter-user angles for which reliable communications between each pair of user-spacecraft can be maintained. This calculation is performed by the The CLASS Automated Conflict Resolution System (ACRS) [1] which takes into account all communication parameters and the antenna pattern in determining the minimum inter-user angle.

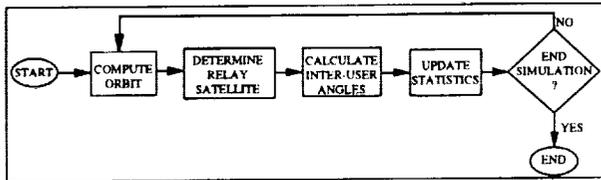


Figure 2. IM Block Diagram

The minimum inter-user angles are then fed into the IM simulation engine. The IM engine performs a point-by-point simulation of each spacecraft's orbit. At each point, the spacecraft's location is determined. IM then assumes that each spacecraft communicates with the nearest visible relay satellite. The inter-user angle between each pair of spacecraft are calculated, and if the inter-user angle is less than the minimum angle computed by ACRS, statistics are recorded. Time is then incremented and the process is repeated.

The orbit generator used by IM was developed specifically for this project and uses a simple geometric model. From the input orbital parameters, the orbit period, the precession rate and the initial orbit are determined. These computed orbital elements are used to calculate the location of the user spacecraft in the orbit plane. Next, the orbit plane is rotated by the inclination angle and spun about the earth's axis at the precession rate, as shown in Figure 3.

IM's orbit generator is designed for speed rather than accuracy so long term statistical data can be calculated quickly. Since it is impossible to predict exact orbits for an extended period of time anyway, the statistical output is sufficiently accurate.

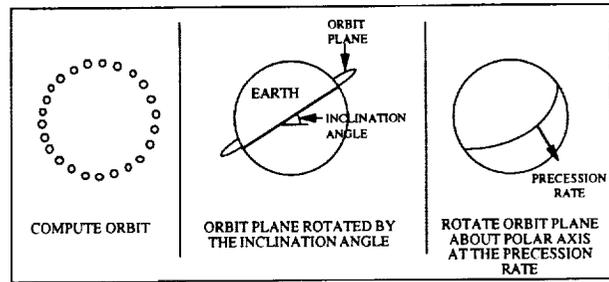


Figure 3. IM Orbit Generator

III. IM CAPABILITIES

IM has many features which make it a versatile tool for evaluating long term mutual interference. In a single simulation, IM can incorporate as many as 100 different user spacecraft, implement several different communication plans and derive mutual interference predictions from years of communication data which has been sampled at increments of time as small as a second. Each communication plan considers a different number of frequency channels and/or different frequency assignments for each user spacecraft. IM can demonstrate when, how much and with whom long-term mutual interference occurs for various communication plans. In addition to the numeric results, IM can reveal where mutual interference occurs utilizing an interactive graphics display.

The ability to incorporate different communication plans in a single simulation make IM a useful tool for frequency planning. The number of communication channels can be varied to create different communication plans within a single simulation. By varying the number of channels, optimal frequency allocations can be identified. Special users can be added or eliminated and their PN coding ability can be turned on or off to determine if PN coding will provide mutual interference isolation.

Mutual interference statistics provided by IM include: the percentage of days with a certain number of minutes of mutual interference, the maximum hours of daily mutual interference and the maximum hours of weekly mutual interference. IM produces mutual interference statistics for each active user spacecraft. Each IM simulation can consider an individual user spacecraft against every other individual spacecraft (see

Figure 4) or against a particular combination of any or all of the other user spacecraft (see Figure 5).

Figures 4 and 5 are statistics from a 25-year run. IM simulated a 25-year period in 1 minute steps and simultaneously gathered statistics for 36 spacecraft. The entire simulation took approximately 3 hours on an HP 9000 model 730 computer. Figure 4 shows the mutual interference between two EOS satellites. It shows that less than 1 percent of the days had mutual interference that was greater than 40 minutes in duration. Figure 5 shows the mutual interference between an EOS satellite and 36 other satellites. Notice that over 1 percent of the days had mutual interference periods greater than 130 minutes.

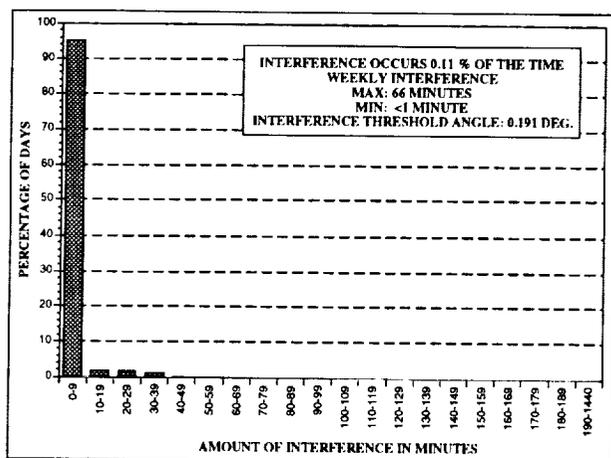


Figure 4. EOS1 vs. EOS2

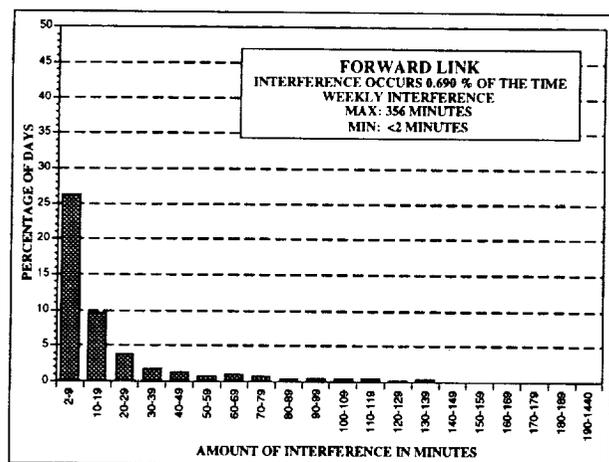


Figure 5. EOS1 vs. All Other Users

IM can also characterize when mutual interference occurs by producing a time-line chart (see Figure 6). The time-line interference chart

displays when coverage loss due to mutual interference occurred over the time-line of a given day. Figure 6 depicts the time-line of a worst mutual interference day. A worst day for a desired user is defined as the day with the smallest number of total contact minutes. Coverage loss due to mutual interference includes any period of mutual interference and any contact period less than 10 minutes. The horizontal bars indicate when mutual interference and Zone of Exclusion (ZOE) outages occurred. The total number of contact minutes and the longest period of time with no contact are also displayed on the chart.

To visualize where mutual interference occurs, the IM interactive graphics mode is available. The interactive graphics mode displays up to three active user spacecraft and the location of mutual interference events on world maps. Maps of the world from the view point of each TDRS satellites are available as well as a flat map of the entire world (see Fig. 7.) Flags are displayed on the map in the location where mutual interference occurred. As the simulation progresses, flags collect and areas that experience the most mutual interference can be identified. At any given time, the IM graphics simulation can be interrupted and communication and orbital parameters of any or all of the user spacecraft altered to determine feasible mutual interference mitigation techniques.

The example shown in Figure 7 is a gray scale print of a color screen. The flags that represent mutual interference make a cross-hatch pattern in-between the continuous sinusoid dotted lines of the orbital paths of the spacecraft (note that the orbital paths of the spacecraft and the flags representing mutual interference are much more apparent when displayed in full color.) In Figure 7, the areas identified with the most mutual interference are immediately before and after the ZOE over the Indian Ocean. Mutual interference is more likely to occur in these areas because the inter-user angles are decreased when user spacecraft are near TDRS horizons.

IV. CONCLUSION

Interference Monitor predicts long term mutual interference statistics between two or more

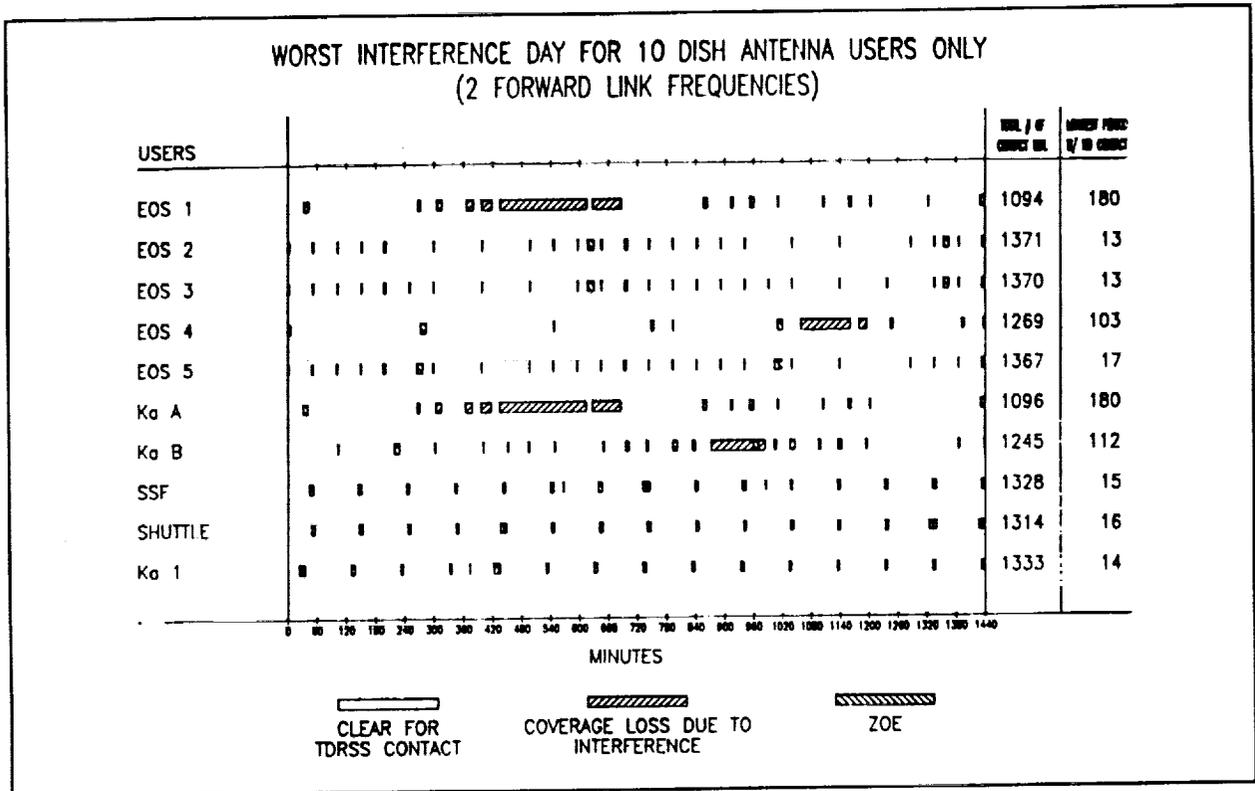


Figure 6. Time-Line for Worst Interference Day

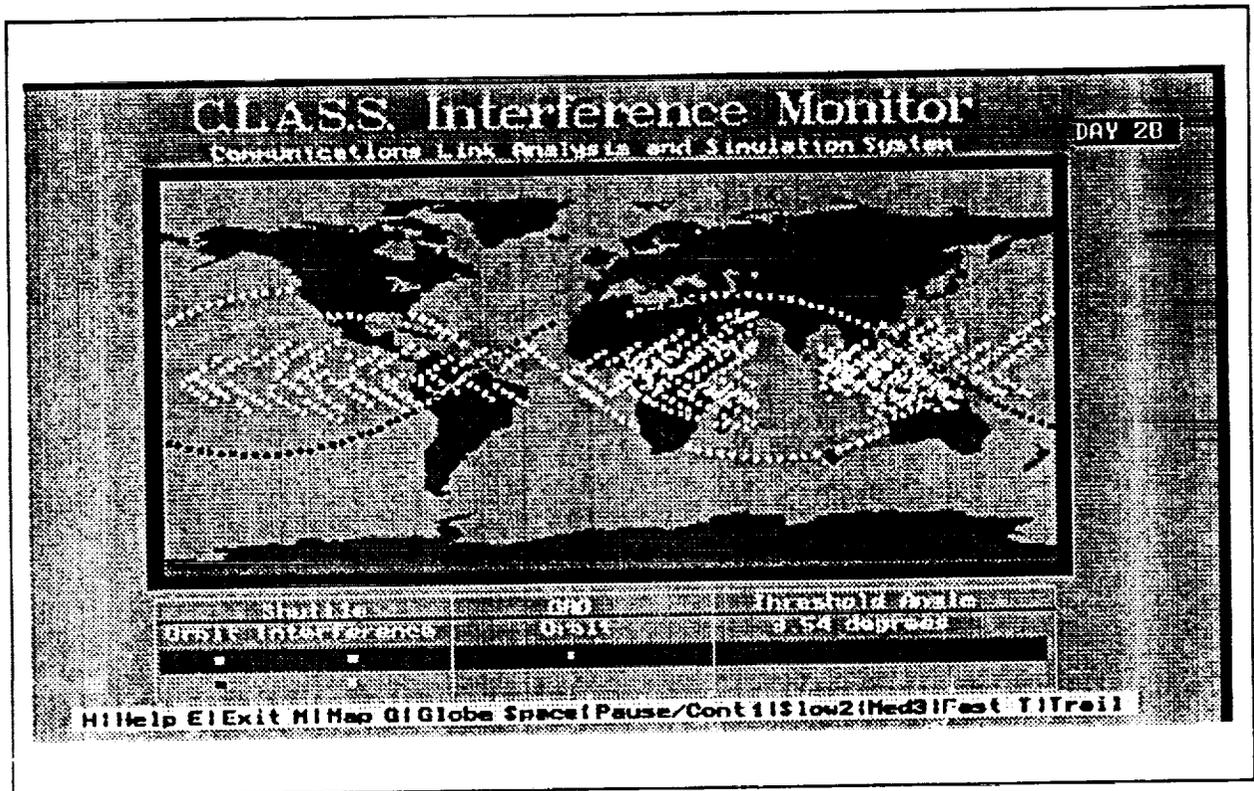
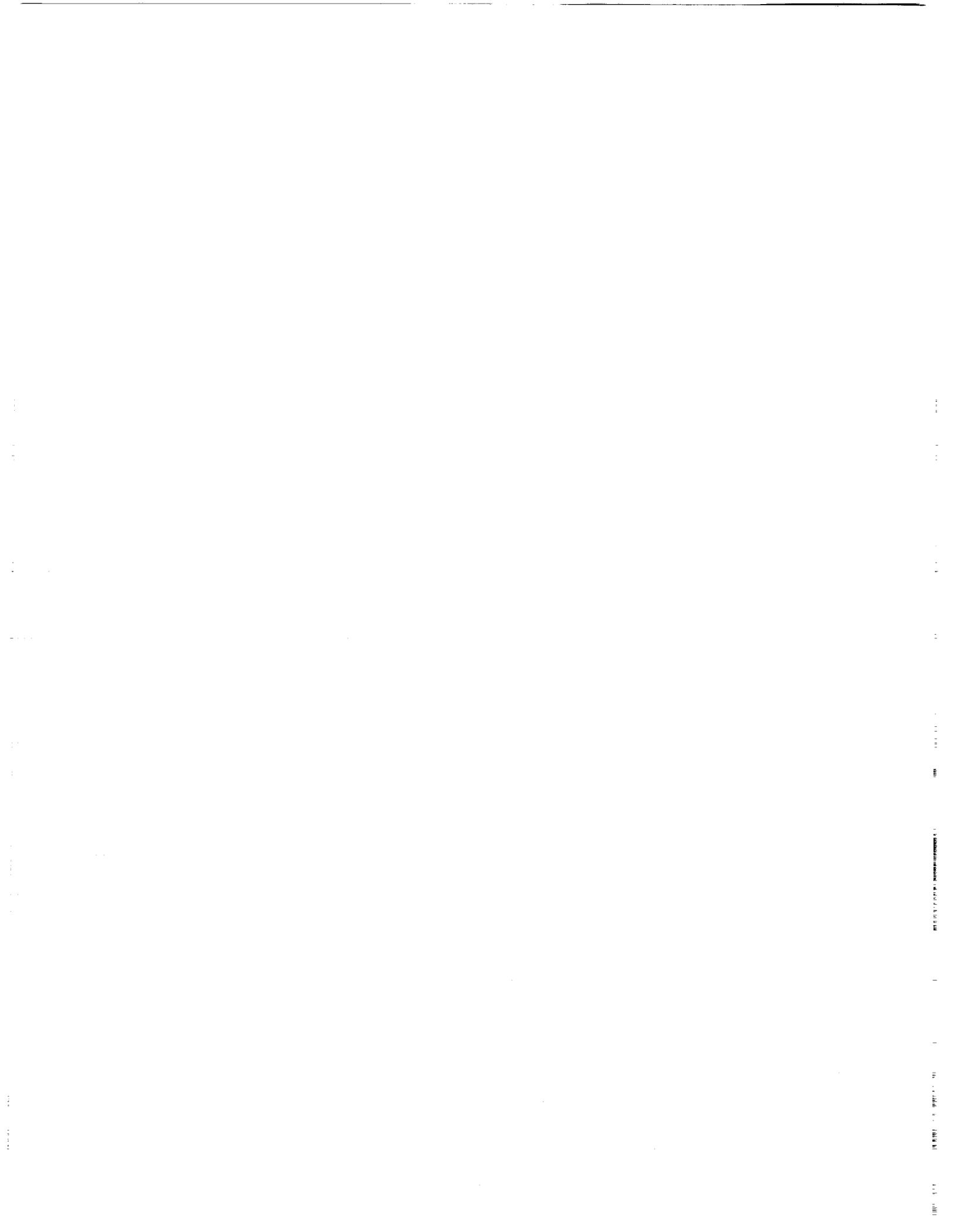


Figure 7. IM Interactive Graphics Screen

spacecraft. IM is used by NASA GSFC for many projects: for network and mission planning and as an aid for both frequency and polarization allocation. NASA headquarters has employed IM for a user spacecraft loading study to help determine network and mission plans for TDRS II user spacecraft. IM is also frequently used by the Space Network Interoperability Panel (SNIP) to study possible interoperability scenarios between the relay satellite systems of NASA, the Japanese Space Agency (NASDA) and the European Space Agency (ESA.) The ease of use and flexibility of IM enables NASA to efficiently determine optimal satellite configurations for the Space Network of the twenty first century.

REFERENCE

- [1] T. Kaplan, J Freedman, D. Wampler, A. Musliner, C. Ruseau, "CLASS Interference Analysis System," Proceedings IEEE National Telesystems Conference, Washington D.C., May 1992, pp. 12-1 - 12-7.



**SESSION 5: TELECOMMUNICATIONS
ENGINEERING**

A PACKET SWITCHED COMMUNICATIONS SYSTEM FOR GRO

Shabu Husain, Wen Hsing Yang and Rani Vadlamudi

AlliedSignal Technical Services Corp.

7515 Mission Drive

Seabrook, MD 20706

Joseph Valenti

Goddard Space Flight Center

Greenbelt, MD 20771

Abstract

This paper describes the packet switched Instrumenters Communication System (ICS) that has been developed for the Command Management Facility at Goddard Space Flight Center (GSFC) to support the Gamma Ray Observatory (GRO) spacecraft. The GRO ICS serves as a vital science data acquisition link to the GRO scientists to initiate commands for their spacecraft instruments. The system is ready to send and receive messages at any time, 24 hours a day and seven days a week. The system is based on X.25 and the International Standard Organization's (ISO) 7-layer Open Systems Interconnection (OSI) protocol model and has client and server components. The components of the GRO ICS are discussed along with how the Communications Subsystem for Interconnection (CSFI) and Network Control Program Packet Switching Interface (NPSI) software are used in the system.

1. INTRODUCTION

The Command Management System (CMS) at NASA-GSFC supports scientific experiments by providing for the planned, safe operation of scientific spacecraft. CMS software is responsible for command request processing, command load generation and checking, constraint checking, automatic command sequence generation, and onboard computer memory management for the spacecraft. One of the primary ground interfaces supported by the CMS is an electronic interface with the remotely-located GRO Instrumenters (NASA/GSFC, [1986]) that allows them to take science data acquisition requests from GRO scientists and translate the requests into command requests. The interface is

essential for GRO science data acquisition as well as the safe and effective control of the GRO spacecraft.

The GRO ICS is based on the International Standard Organization's 7-layer Open Systems Interconnection (OSI) protocol model. A schematic representation of the ISO-OSI model is shown in Figure 1.

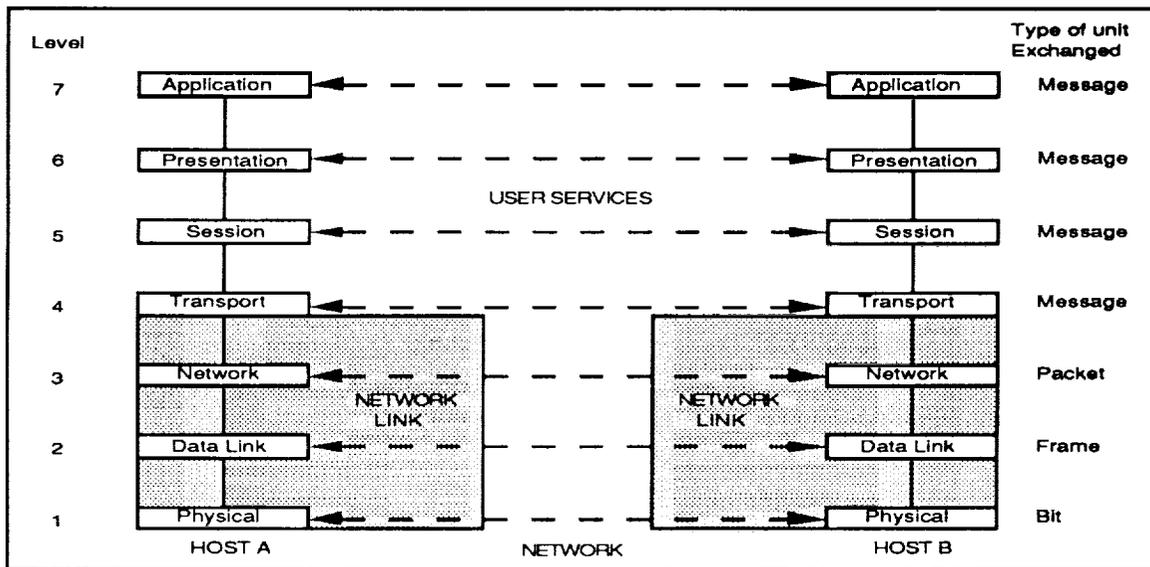


Figure 1. Representation of the ISO-OSI Model for Network Architecture

Each level within the OSI model performs a specific function while allowing the systems being connected to have a heterogeneous nature. The functions of levels 1 to 3 of the model deal with the internal mechanisms of the network and their interfaces to the hosts. This is significant for X.25 because the highest level of protocol that X.25 defines is the packet level (or X.25 at Level 3) which takes care of all OSI network (Level 3) layer functions and some transport layer (Level 4) functions. Levels 4 and above deal with protocols for controlling processes on the hosts themselves.

The first three (bottom) levels of the GRO ICS were implemented using X.25. These levels control the exchange of data between the user devices and the packet network node of the packet switching network. At the physical layer (layer 1), the NASA Communication (NASCOM) Division provides dedicated communication services to accommodate data transfer between the CMS and the users. The top four layers were developed and integrated for the GRO ICS in accordance with the ISO model recommendations.

The GRO ICS has been designed and built with the client/server concept to provide each of the four remotely-located GRO Instrumenters with an X.25 interface that is able to send and receive messages at any time, 24 hours a day and seven days a week. For incoming messages from any GRO Instrumenter the ICS functions as a server to store and forward the messages to the CMS for further processing. For outgoing messages from the CMS to the investigators, the ICS functions as a client to establish a communication link to the designated instrumenter's communication server task. In addition to client and server components, the GRO ICS includes the Application Programming Interface (API) component and the Network Subsystem Software (NSS) component. These components are shown schematically in Figure 2. Each of the components is described in more detail in the following sections of this paper.

2. THE SERVER

The GRO ICS has four identical servers, one for each GRO Instrumenter, and each server has a subserver. Each server controls the attaching, detaching and scheduling tasks of the GRO ICS and maintains the sessions which are in progress. Certain tasks are attached by the server at the start up time while others are attached as they are required. At the start up, the server task attaches LSTNR1S, LSTNR2S and one standby receiver task (SUBSRV01). If the server receives a ring request from an investigator, it posts the standby receiver task (SUBSRV01) and

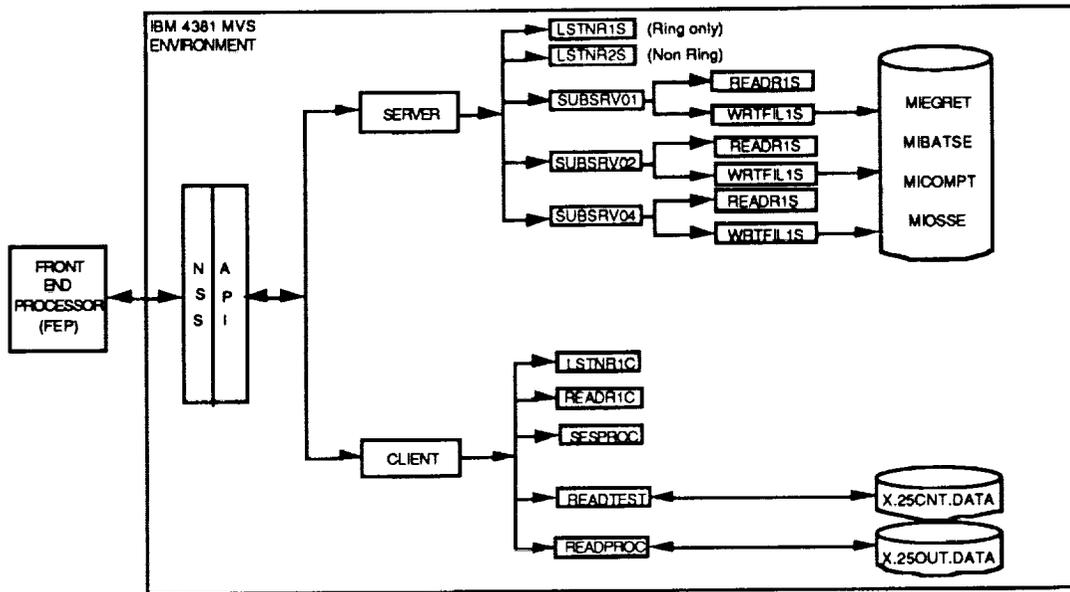


Figure 2. GRO ICS Structure Diagram

attaches a new standby receiver task (SUBSRV02). For every new ring request the standby receiver task is posted and a new receiver task is attached in the standby mode. This procedure is faster than attaching a receiver task after the receipt of a ring request.

The LSTNR1S and LSTNR2S tasks are similar to each other, except that they listen for different types of messages. LSTNR1S listens for supervisory messages (ring-only requests) coming over LCN0. At the same time, LSTNR2S listens for all non-supervisory messages on LCN0. LSTNR1S is scheduled to be executed at the highest priority among all the subtasks of the server, which ensures the quickest possible reply to incoming ring requests.

As soon as a message arrives at either the LSTNR1S or LSTNR2S tasks, the server task is posted and the content of the message is deciphered to determine whether it is Logical Channel Number (LCN) oriented or a non-LCN oriented. If the message is LCN oriented, the server determines which of the attached tasks is assigned to that particular LCN. If the message is non-LCN

oriented, the server posts each and every read task as well as the write task. Each task will in turn process the message.

After determining the responsible task for a supervisory message, the server posts the task and action is taken by either the READR1S or the WRTFIL1S task. The function of READR1S task is to read all the data coming from the FEP for the specified LCN value. The function of the WRTFIL1S task is to write the incoming data messages to one of the four user files (MIEGRET, MIBATSE, MICOMPTTEL or MIOSSE), depending on who initiated that session. WRTFIL1S is created so that the processes of writing data to a file and receiving data messages can be done asynchronously.

3. THE CLIENT

The client (sender) task has some similarity to the receiver task (SUBSRV). The client attaches two tasks (LSTNR1C and READR1C) during initialization. The function of the LSTNR1C task is to listen for incoming replies over the specified LCN from the FEP. The SESPROC task builds the session control blocks as the session handshaking requests are received from the session control file (X25CNT.DATA). The client then initiates a session by sending a connect request for an available LCN and a ring request to the appropriate remote host. After a ring has been accepted, the client initiates the session initiation handshake. If the above steps are successful, the client sends the message from the output queue (X25OUT.DATA) to the appropriate remotely-located instrumenter.

After successfully sending data to the remotely-located instrumenter, a session termination request is sent to the host followed by a clear request to terminate the connection. The client task will perform these steps every time an automatic or manual session handshaking request is received from the operator. The advantage of initiating an automatic session handshake is

that sending and receiving session control blocks and transmitting data is accomplished without any delays or session sequencing errors. When there is no communication activity, the client can be terminated gracefully so that only the server's listener task is active. With this design for the client component, the overhead on the GRO ICS operating system is significantly reduced.

4. THE APPLICATION PROGRAMMING INTERFACE (API)

The API provides users with access to the X.25 networking capability and the API application programs that are running under the IBM MVS operating system. Several C-language application programs were written for the API to access the NSS.

5. THE NETWORK SUBSYSTEM SOFTWARE (NSS)

The NSS acts as a message processor for requests for network services from the API application programs. The GRO Instrumenters are able to establish connections, send and receive data, and terminate connections via requests that are passed by the NSS to the network. The NSS multiplexes requests from users through I/O sub channels while it manages all of the queues and control blocks necessary to support network traffic. The NSS is interfaced to the Comten 3695 Front End Processor (FEP) which has an X.25 interface to the IBM 4381 mainframe that supports the CMS.

6. CSFI AND NPSI SOFTWARE

The GRO ICS currently uses IBM Communications Subsystem for Interconnection (CSFI) software. CSFI is a communication and transmission control program for networking IBM computers with non-SNA hosts through an X.25 network. CSFI was developed by IBM in France as the Generalized Transaction Monitor for Open System Interconnection (GTMO SI) software. The product became available

as CSFI in Europe and Canada in April, 1990. GSFC was able to obtain CSFI in Dec. 1990, a month before CSFI was officially released in the U.S. This allowed the GRO ICS to be the first system in the U.S. to use CSFI software under the MVS operating system.

CSFI allows state-of-the-art communication system applications to be written through a programming interface. The programming interface consists of a set of macro statements that can be invoked in an assembler language program. CSFI also provides ready-made services for frequently used types of connections between heterogeneous elements. Some of these services may be customized to a user's unique requirements. In addition, CSFI provides the user with the ability to process X.25 call packets.

When a call packet is received from a GRO Instrumenter, CSFI sends out a call accept packet and starts the server task to receive data. The server calls a subserver that creates a queue to store any data received from the instrumenter. A timer is started and the server waits for a message from the instrumenter. If a session initiation request is received, the server sends a session initiation acceptance message and waits for the next message. If a message initiation request is received, the server sends a message initiation acceptance message and waits for the next message. If data is received, it is written to the queue and the server waits for another message. If a message termination request is received, the server sends a message termination confirmation message and waits for the next message. If a session termination request is received, the server sends a session termination confirmation message, waits for the primary to terminate, and passes control to the subserver.

When there is data, a subserver is called by the server. A sequential queue is created to receive the data, control is passed back to the server, and the server waits for the subserver to terminate. Once the subserver has terminated, an entry from

the queue is read. If the entry is a message header, bytes 4 through 32 are translated to EBCDIC and the remaining part of the buffer is filled with null characters. The message name is retrieved from the header, and the header is written to one of four user files (MIEGRET, MIBATSE, MICOMPTEL and MIOSSE), depending on who initiated the session. If the entry consists of data, bytes 4 through the end of the record are translated to EBCDIC and written to a dataset. If the entry is a message trailer, bytes 4 through 32 are translated to EBCDIC and written to the dataset. Finally, the dataset is closed and the event is queued to the monitor. These steps are repeated for each event until the queue is empty.

A client transaction is started by CSFI when a call packet is to be sent out to the GRO Instrumenter. The files to be sent are read from the X25CNT.DATA file using the READCNT transaction. Based on the destination, the proper connection is established using the CONN macro. A session initiation request packet is built and a session initiation request message is sent to the secondary partner. When a message is received from the secondary partner, the message is deciphered. If the message block received is a session initiation acceptance block, the client sends a message initiation request message and waits for the next message. If a message initiation acceptance message is received, the client sends the data packets. After the data packets are received by the secondary partner, the client sends a message termination request message and waits for the next message. If a termination confirmation message is received, the client sends a session termination request message and waits for the next message. If a session termination confirmation message is received, the session with the secondary partner is ended.

In addition to the CSFI software, the current GRO ICS uses the X.25 Network Control Program Packet Switching Interface (NPSI) software from NCR. NPSI provides an interface for Systems Network Architecture (SNA) users to use X.25 Packet Switched Data

Networks (PSDNs) in conjunction with their existing networks. This interface allows SNA host processors to communicate with both SNA and non-SNA equipment over PSDNs that use X.25 protocols. The NPSI software has General Access to X.25 Transport Extension (GATE) features that enable a host application program to completely control the establishment and clearing of X.25 virtual circuits. With this software, the GRO ICS is able to control the operation of the packet level protocol on individual virtual circuits, the operation of the X.25 interface for including incoming and outgoing calls, and the status of the link.

7. CONCLUSIONS

The GRO ICS provides a vital electronic link between the Command Management Facility at GSFC and the remotely-located GRO Instrumenters. The successful design of the GRO ICS depends on the ISO-OSI 7-layer protocol model, X.25, the client/server concept, packet switching technology, and commercial CSFI and NPSI software. The GRO ICS was the first system in the United States to use the IBM CSFI software under the MVS operating system. The NCR NPSI software with GATE features was very helpful for connecting the SNA host with non-SNA hosts on the X.25 packet switched data network.

ACKNOWLEDGMENTS

The authors would like to thank Patricia Lightfoot, Mike Paoletta and Alan Johns at NASA-GSFC/Code 514 and Ellen Stolarik, David Warren, Vijaya Sonti and Ronald Littlefield at AlliedSignal Technical Services Corporation for their continual support and many contributions to this communications task. This communications task is supported by NASA contract #NAS5-27772.

REFERENCES

1. NASA/GSFC, Interface Control Document between the Packet Processor (PACOR) and the Command Management System and the PACOR and CMS Users, Goddard Space Flight Center, Greenbelt, MD, Nov. 1986.

USING THE USCCS FOR SUB MICROSECOND SPACECRAFT CLOCK CALIBRATION

Victor J. Sank, Ph.D.
Fredrick Herold & Associates, Inc.
7303A Hanover Parkway
Greenbelt, Maryland 20770
(301) 982-4001

Summary

The Return Data Delay technique which requires knowledge of spacecraft range is commonly used for correlating a spacecraft clock against a ground time standard when millisecond accuracy is required. An analysis is presented that allows using the user spacecraft clock calibration system (USCCS) to correlate a spacecraft clock to better than one microsecond accuracy. The basic USCCS algorithm has been simplified and it is shown to result in about one microsecond accuracy without requiring orbital information. By considering the relative motion of the user satellite, the TDRS and the earth station about the center of the earth, a correction of almost two orders of magnitude can be made. Such accuracy is required for scientific investigations that require correlating coincident radiation or particle detection with a remote laboratory.

Background

An accurate absolute time reference is required on scientific spacecraft for operational purposes and for correlating scientific data. Typical accuracy required for many satellites has been of the order of one millisecond; examples are ERBS, SMM, SME, UARS, EUVE, LANDSAT and HST. This accuracy is obtained by reading the spacecraft clock (or oscillator count values) simultaneous with a particular bit edge in the telemetry. As the telemetry is received on the ground an absolute time reference is recorded periodically for selected bits in the telemetry. The absolute ground receipt time of the bit edge that was simultaneous with the onboard clock reading can thus be determined. Orbital knowledge and equipment time delays then allow the calculation of the time the particular bit was created on board and, hence, the time the spacecraft clock was read based on the ground clock. The reading on the spacecraft clock is sent

to the ground in a later portion of the telemetry and serves as a clock calibration by comparing it with the earth based standard.

The main limitation in this method, which is referred to as the return data delay (RDD) or return channel telemetry delay (RCTD), is accurate knowledge of the spacecraft range (and thus the space propagator delay) from the ground receiving station. The calculations are done after a pass with the tracking, data and relay satellite system (TDRSS) and uses predicted orbital positions. The results are extremely sensitive to the quality of the orbital data but can be accurate to about 3 μ s with respect to the ground time reference (atomic clock at the NASA Ground Terminal (NGT)). The RDD method is an open loop method.

The CGRO project decided that because they required a 10 μ s clock setting accuracy, they would take advantage of the pulses used in the TDRSS spacecraft ranging system and implement a closed loop time correlation technique. This became the user spacecraft clock calibration system (USCCS). GPS accuracy is approximately 1 μ s and today would be a competing technique.

Using Range Pulses for Time Correlation

The standard ranging technique used in the TDRSS is a closed loop method in which a pulse is sent from the TDRSS ground station at White Sands New Mexico (WSGT) to a user via a TDRS and then returned from the user to the WSGT via the same TDRS. The signal propagation time which is used to determine the range to the user is combined with the known TDRS orbit to determine the user satellite orbit. The time accuracy between pulses is specified to be \pm 35 nanoseconds. For orbit determination, doppler shift is also used.

The changes required to use the range pulses for timing were:

1. to extract the range pulse from the spacecraft transponder and use it to trigger a reading of the spacecraft clock.

2. to make an absolute time reading of the ground station range pulse transmission time, and the range pulse receipt time. For range purposes the delta between these times were measured accurately but the absolute times were not.

The range pulses, referred to as pseudo random (PN) code epoch pulses, are the part of the PN spectrum spreading code used in the TDRSS where there are 18 ones in a row. The code has a length of 256 x 1023 chips (PN bits) generated at a rate of approximately 3MHz. This results in a PN epoch pulse approximately every 0.085 milliseconds (25500 Km at the speed of light).

USCCS

The basic concept of the USCCS time correlation method is extremely simple. When a PN epoch pulse is transmitted from the ground the time is recorded, t_1 . It arrives at a spacecraft at some time t_2 which is unknown. The spacecraft transponder immediately retransmits the pulse to the ground where its arrival time, t_3 , is recorded. Upon receipt of the pulse at the spacecraft, the spacecraft's clock is read and the reading, t_{2sc} , is transmitted to the ground along with other telemetry. On the ground, t_2 is calculated from t_1 and t_3 and compared to t_{2sc} for calibration. The pulse arrival time at the spacecraft t_2 is approximately half-way between t_1 and t_3 .

$$t_2 = \frac{t_1 + t_3}{2} \quad (1)$$

When the signal propagation is viewed by an observer on the earth it may seem that this is not an approximation. Proper analysis, however, requires that the system be viewed from an inertial reference frame.

For a low earth orbiting (LEO) spacecraft at about a 400 Km altitude, and a geosynchronous relay satellite, the round trip distance traveled by an epoch pulse never varies by as much as 25500 Km, thus, there is no ambiguity between a transmitted epoch and its corresponding return epoch. No orbital information is needed, other than t_1 and t_3 , when correlating epochs to approximately determine t_2 .

Accuracy of the USCCS

The correction to half-way between t_1 and t_3 does not depend on the LEO spacecraft velocity or the RF signal doppler shift. The correction does depend on the LEO position and the fixed quantities: earth rotational speed; geosynchronous relay satellite rotational speed; relative position of earth station; and, of course, the speed of light. The solution for t_2 with the various motions is now examined and numerical values are given for an example of orbits in the earth equatorial plane. It will be shown that the required correction of up to about $1 \mu s$ is asymmetric with the relative position of the LEO and relay satellite and, thus, not related to the doppler which is almost symmetric. The doppler can be used if orbital data is not otherwise available to determine the longitudinal component of the relative ground station, TDRS, and user position, which can then be used to determine the required correction to Equation 1.

Doppler

Consider the example of an object (user spacecraft) moving away from a fixed observer at a constant velocity, Figure 2. The frequency transmitted from the observer, f_1 , and received by the observer, f_3 , are different (doppler), BUT the transmitted epoch pulse is at the user at a point in time t_2 , which is exactly half-way between when it leaves the observer, t_1 , and returns to the observer, t_3 . Note that the relative velocity, v , between

observer and user spacecraft can be determined from the observer's measurement of forward and return PN periods, T_1 and T_3 , respectively. For the USCCS, an observer at the WSGT on a slowly rotating earth and a spacecraft in orbit, the above concept is approximately true. For the constant velocity case this equation is exactly true no matter how fast the user spacecraft is moving.

A simulations viewing earth, TDRS, and user spacecraft actual motion from a reference frame moving with the center of the earth shows the correction term to Equation 1 to be less than $1 \mu\text{s}$ and, again, independent of the user spacecraft motion, Figure 3. This analysis thus tells us that Equation 1 is accurate to $1 \mu\text{s}$.

Special relativistic considerations are not included since they are of the order v^2/c^2 and which is about one nanosecond ($v^2/c^2 \times$ round trip time of 0.5 second).

Simulation

We observe the three bodies from above the north pole of the earth with the earth center being the center of our reference frame, and the TDRS and user orbit in the same plane, Figure 3. Due to the earth's curved path (orbit) around the sun this also is not a true inertial reference frame, but the remaining errors are below our desired accuracy.

From Figure 3, we see that the forward path and return path are not the same. Since the forward and return paths are different, the signal that left the ground at t_1 and returned at t_3 is not at the spacecraft exactly half-way in between. Since the signal is at the user for only an instant of time at t_2 , it may be intuitive that the difference in forward and return time is independent of the user spacecraft velocity. This is similar to the constant velocity example given earlier.

The difference in forward and return travel time is due to both the earth and TDRS motion. Clearly the TDRS motion during the 0.25 second that the signal travels from the TDRS to user and

back to TDRS causes a difference in forward and return travel time. In addition, a more subtle effect is the difference between the GT-to-TDRS and TDRS-to-GT travel time. As a signal moves through space from the GT to TDRS-E, the TDRS is moving away. Thus, the distance traveled by the signal is greater than the instantaneous GT-to-TDRS distance. Similarly, as a signal travels from TDRS-E to the GT, the GT moves toward the signal so the distance traveled by the signal is less than the instantaneous distance. Since the TDRS and GT are always in fixed relative positions, the forward travel time of one pulse is the same as the forward travel time of the following pulse. Similarly, the return travel time from TDRS to GT for successive pulses is the same. For TDRS-E the forward time is $.395 \mu\text{s}$ longer than the return time. For TDRS-W the values are reversed.

We now consider the portion of the signal path between the TDRS and the user. The computer simulation shows the signal travel time from TDRS to the user to be a maximum of $0.466 \mu\text{s}$ longer than the travel time from the user to the TDRS. This occurs when the angle between the TDRS and the user spacecraft is approximately 90° , the LOS point. At AOS the values are reversed because the user is approaching the TDRS and the TDRS-to-user time is less than the user-to-TDRS portion. For a user in a polar orbit, the effect is less.

The net effect of combining the GT-TDRS and TDRS-user portions is such that for TDRS-E at LOS the forward travel time is $0.86 \mu\text{s}$ longer than the return travel time. At AOS it is about $0.072 \mu\text{s}$ less, Figure 4. For TDRS-W the conditions are reversed.

As in the constant velocity example of Figure 2, the forward and return PN periods T_1 and T_2 can be used to determine the longitudinal component of the user's velocity, v . In the real case the complete motion, as shown in Figure 3, must be taken into account, but the results will be approximately equal to that given by the equation for v in Figure 2.

The maximum doppler effect will cause the PN period at the spacecraft to vary by about $2 \mu\text{s}$ from the period transmitted from

the GT. Thus, the return PN period at the GT will be a maximum of 4 μ s different from the transmitted. Even though the PN periods as measured at the GT reflect the longitudinal component of the velocity of the user spacecraft, Figure 5, their magnitude does not directly relate to the correction needed in the USCCS to determine t_2 . The values of T_1 and T_3 serve only to tell us where the user is in his orbit relative to TDRS. The geometry of Figure 3, the TDRS motion and earth rotation, must then be used to calculate the sub-microsecond correction needed to obtain the correct value of t_2 . That is, T_1 and T_3 at a given time are used to find v . Knowing v we can find the relative orbital angle between TDRS and the user, Figure 5. Once the angle is known, the difference between the forward and return travel time, $t_F - t_R$, can be found from Figure 4 and used to find the correction required for Equation 1.

$$t_2 = \frac{t_1 + t_3}{2} + \frac{t_F - t_R}{2} \quad (2)$$

Correction Term

Using Figures 3, 4 and 5, we can model the correction term $t_F - t_R$ as follows. The user velocity as seen from TDRS (not orbital velocity) varies approximately sinusoidally as

$$v = v_u \sin \frac{\pi}{2} \frac{\theta_r}{\theta_c}, \quad -100^\circ \leq \theta_r \leq 100^\circ \quad (3)$$

where:

θ_r is the relative angle between the TDRS and user measured from earth center,

θ_c is $\cos^{-1} \frac{r_u}{r_T}$, the relative angle at which the user is moving directly toward or away from TDRS,

v_u is the user's orbital speed.

In terms of T_1 and T_3 , v is approximately

$$v = \frac{T_3 - T_1}{T_1 + T_3} c \quad (4)$$

Equation 3 is used to solve for θ_r ,

$$\theta_r = \frac{2\theta_c}{\pi} \sin^{-1} \frac{v}{v_u} \quad (5)$$

where v is known in terms of T_1 and T_3 from Equation 4.

From Figure 4, the correction term $t_F - t_R$ can be represented by

$$t_F - t_R = .4665 \sin \theta_r + .3945 \mu s \quad (6)$$

Using Equation 2 with this correction yields about two orders of magnitude correction over Equation 1. T_1 and T_3 are sufficiently slowly varying that $T_1(t_1)$ and $T_3(t_3)$ will suffice.

Conclusion

Exploring the physics of the USCCS for the CGRO because of a problem that was found to be simply a misplacement of data in the telemetry, lead to the previous analysis. After examining the full signal path from an inertial point of view it became clear that the basic formula, $t_2 = (t_1 + t_3)/2$, is accurate to $1 \mu s$ without any further correction (except of course, equipment delays). The analysis did show that greater accuracy, approaching that of the TDRSS ranging system, is possible but requires knowledge of the relative position of the user spacecraft and ground station relative to the TDRS. Other limitations in the system as currently implemented are that the time tags are only resolved to $0.1 \mu s$; and, although sub microsecond accuracy with respect to the ground station time standard is possible, it is only kept within about $1 \mu s$ of UTC.

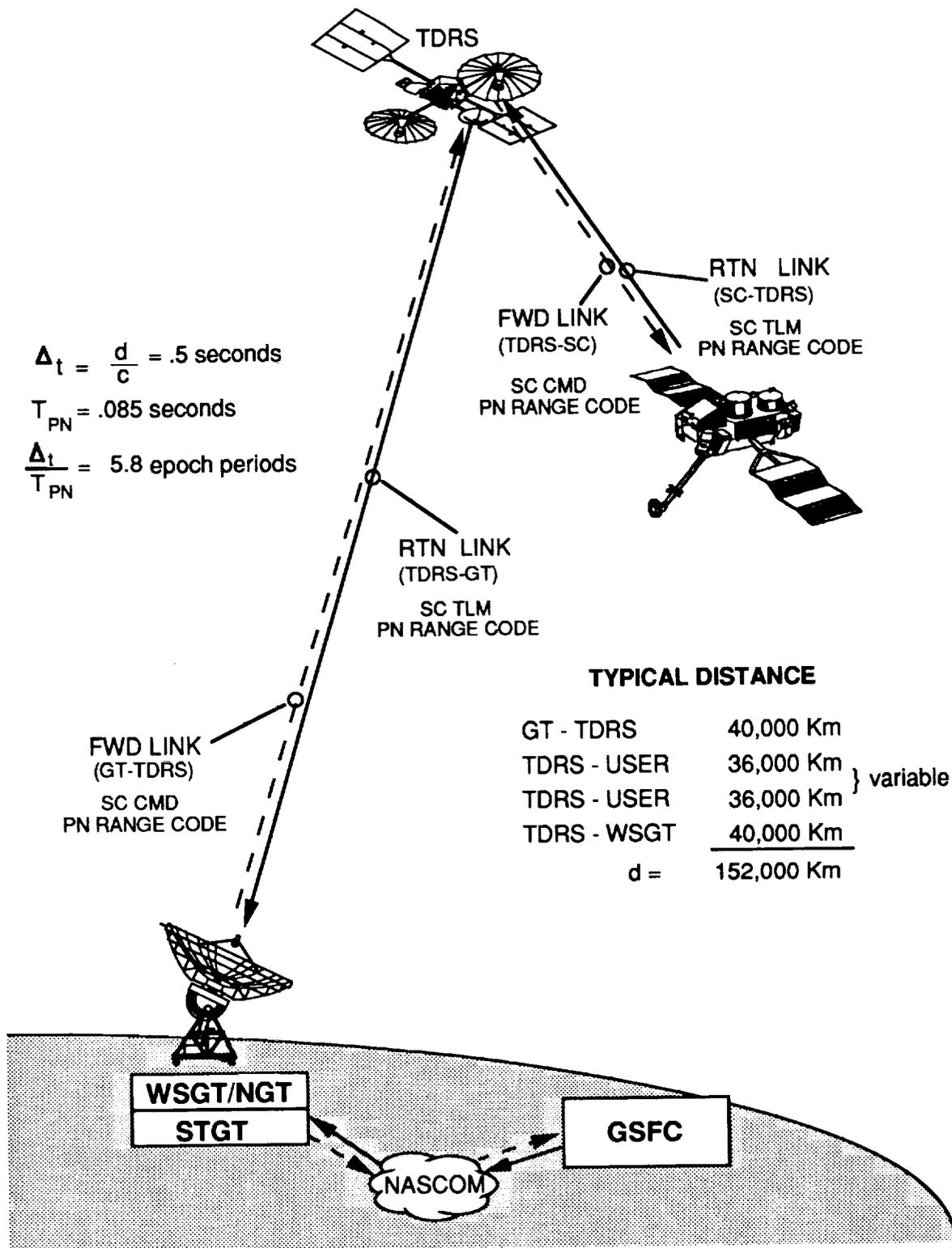


Figure 1. System Overview

v = constant
 T_1 = FWD PN period at GT
 T_2 = PN period at SC
 T_3 = RTN PN period at GT
 λ_1 = FWD PN wavelength at GT
 λ_2 = PN wavelength at SC
 λ_3 = RTN PN wavelength at GT
 c = speed of light

$$\frac{\lambda}{T} = c$$

$$\lambda_2 = \lambda_1 - v T_2$$

$$T_2 = \frac{c}{c-v} T_1$$

$$t_2 = \frac{t_1 + t_3}{2}$$

$$\lambda_3 = T_2 c + T_2 v$$

$$T_3 = \frac{v+c}{c} T_2$$

$$T_3 = \frac{v+c}{c-v} T_1$$

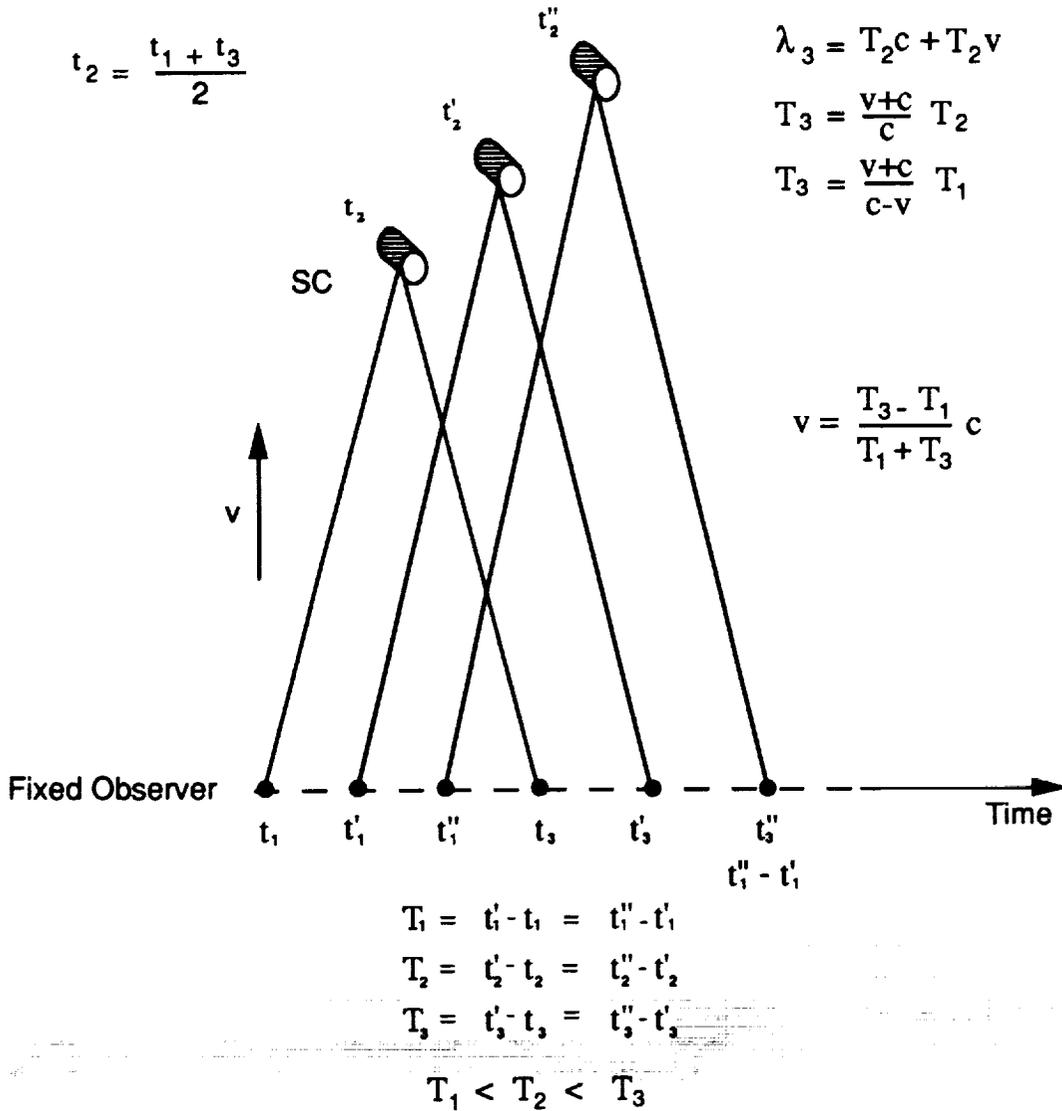


Figure 2. Signal Path For Object At Constant Velocity

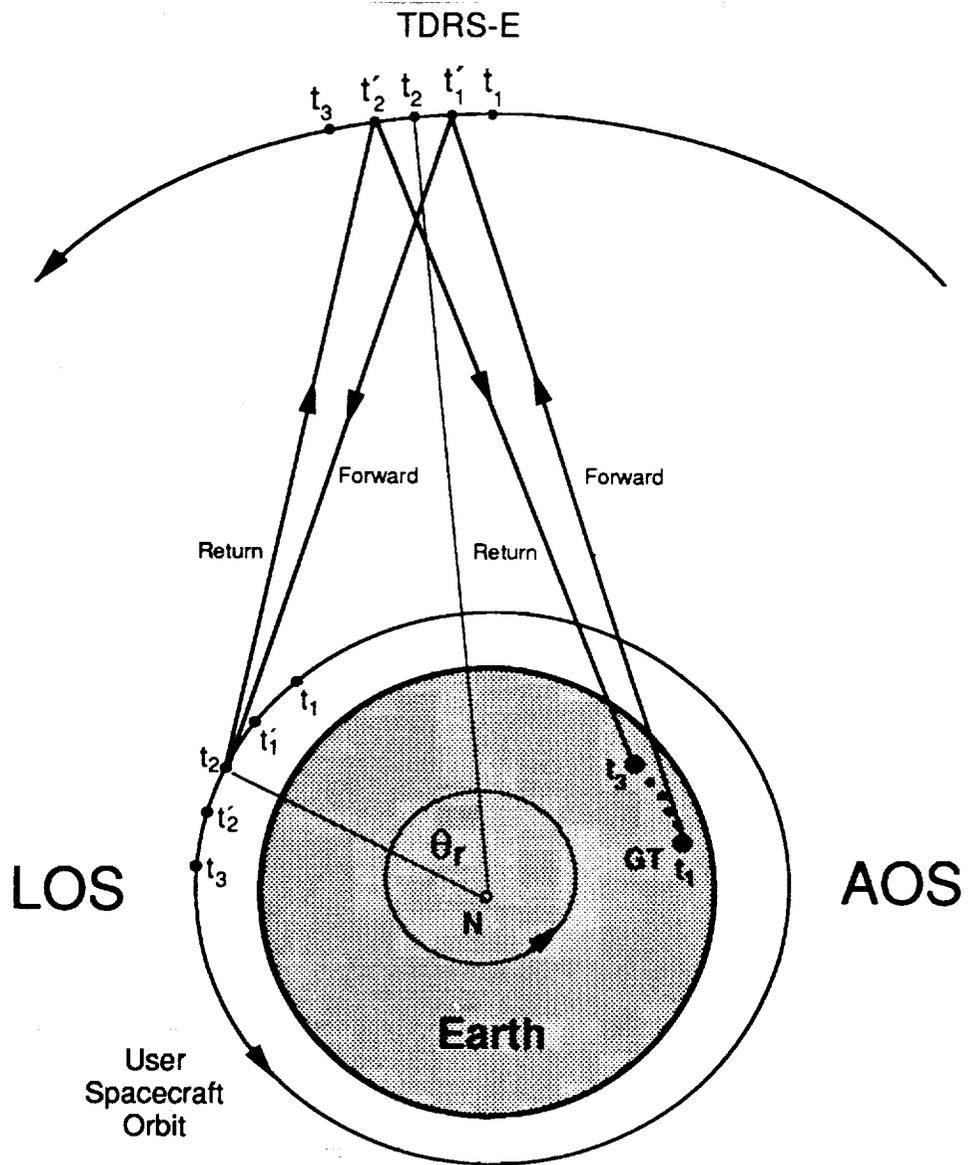


Figure 3. RF Signal Path

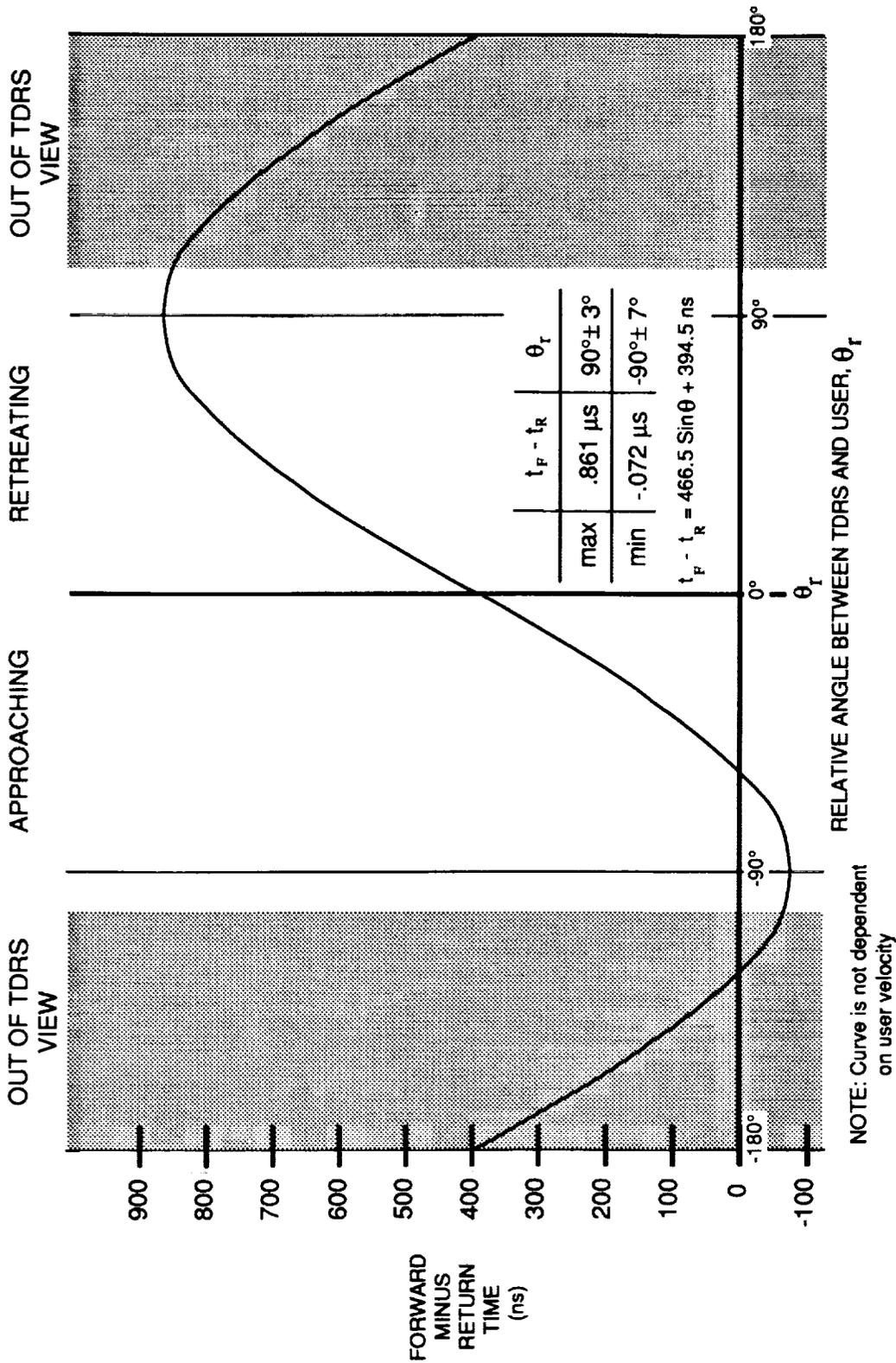


Figure 4. Difference Between Forward And Return Signal Travel Time ($t_F - t_R$)

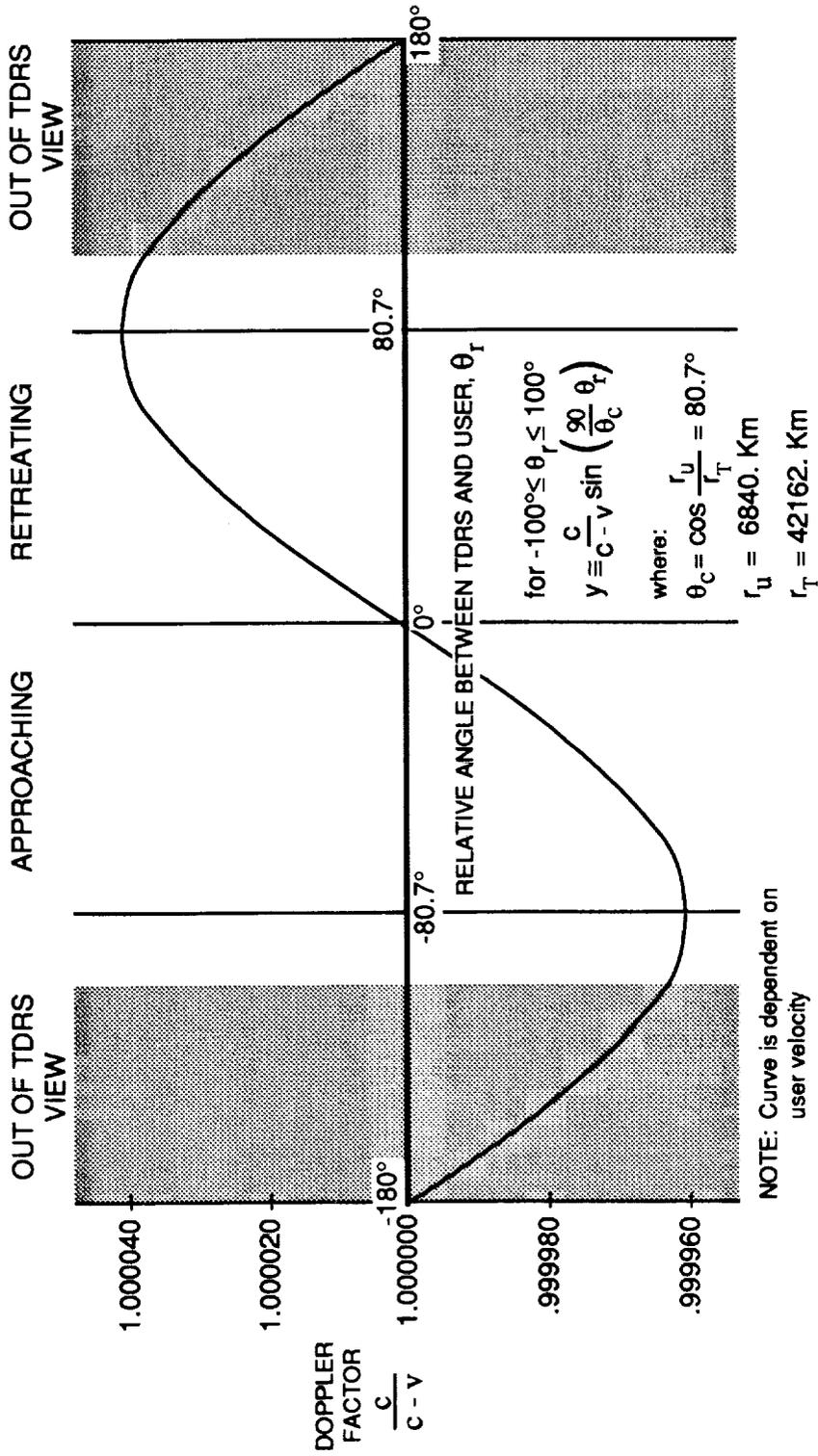
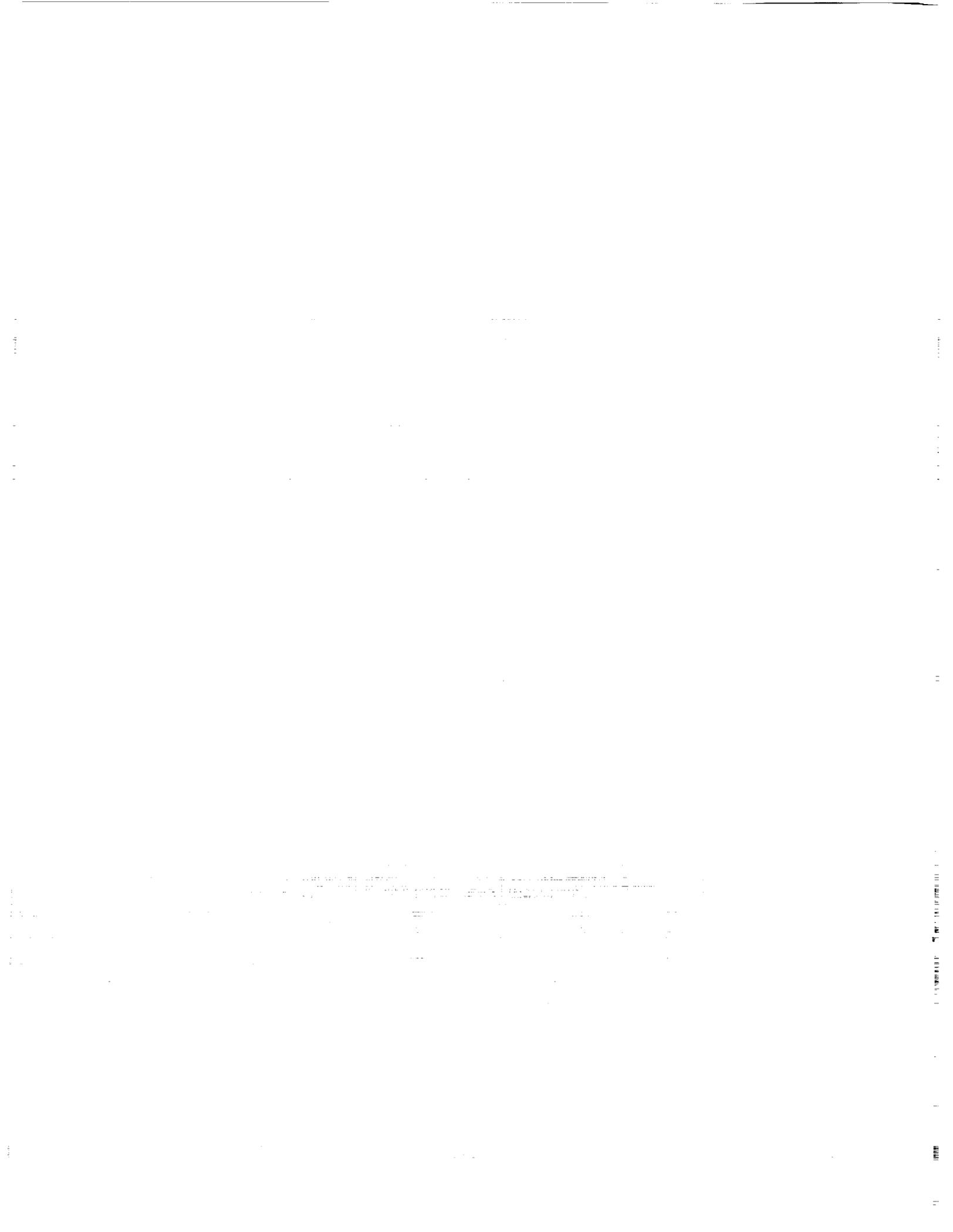


Figure 5. Doppler Factor $\frac{c}{c-v}$



**A LOW COST ALTERNATIVE TO HIGH PERFORMANCE
PCM BIT SYNCHRONIZERS**

Bruce DeShong

AlliedSignal Technical Services Corporation, ATSC

Data Systems Group, DXG

Design and Development Section, DDS

Columbia, Maryland 21045

SUMMARY

The Code Converter/Clock Regenerator (CCCR) provides a low-cost alternative to high-performance Pulse Code Modulation (PCM) bit synchronizers in environments with a large Signal-to-Noise Ratio (SNR). In many applications, the CCCR can be used in place of PCM bit synchronizers at about one fifth the cost. The CCCR operates at rates from 10 bps to 2.5 Mbps and performs PCM code conversion and clock regeneration. The CCCR has been integrated into a stand-alone system configurable from one to six channels and has also been designed for use in VMEbus compatible systems.

PRECEDING PAGE BLANK NOT FILMED

A LOW COST ALTERNATIVE TO HIGH PERFORMANCE PCM BIT SYNCHRONIZERS

Bruce DeShong

AlliedSignal Technical Services Corporation, ATSC

Data Systems Group, DXG

Design and Development Section, DDS

Columbia, Maryland 21045

ABSTRACT

The Code Converter/Clock Regenerator (CCCR) provides a low-cost alternative to high-performance Pulse Code Modulation (PCM) bit synchronizers in environments with a large Signal-to-Noise Ratio (SNR). In many applications, the CCCR can be used in place of PCM bit synchronizers at about one fifth the cost. The CCCR operates at rates from 10 bps to 2.5 Mbps and performs PCM code conversion and clock regeneration. The CCCR has been integrated into a stand-alone system configurable from one to six channels and has also been designed for use in VMEbus compatible systems. This paper compares the functions and performance of the CCCR to those of the higher-cost PCM bit synchronizers and describes typical applications of each device as well as the use of the CCCR to reduce system costs at the Merritt Island (MIL) Tracking Station.

1. BACKGROUND

The primary functions of the CCCR and PCM bit synchronizer are identical; however, each device has been designed to operate in different environments. To provide a clearer identification of the differences between these units, some background information is presented on their basic functions.

Pulse Code Modulation

Applicability. This discussion of PCM is limited to the encoding scheme as it applies to standardized binary/digital symbol based systems.

PCM. The CCCR and PCM bit synchronizer perform functions in support of the serial transmission of digital data. *Pulse Code Modulation (PCM)* is a method commonly employed in this form of data communication. For the purpose of understanding PCM, consider a simple communication system consisting of a data source and a data receiver connected via a single conductor, as shown in Figure 1.

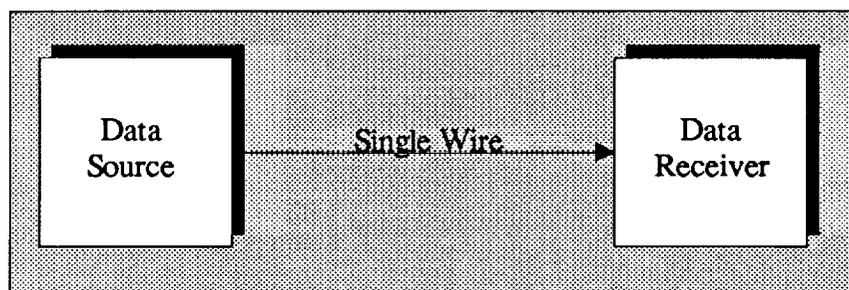


Figure 1. Communications System

For a binary system, PCM defines a set of simple rules governing the transmission of digital data (i.e., a logical "one" or logical "zero") from the source to the receiver by varying the potential on the wire over a period of time referred to as the *bit period*.

Although there are a number of different standardized binary PCM coding schemes, all are based on two simple characteristics of the signal being transmitted: voltage levels and voltage transitions. Assume that the potential in the wire can only assume one of two discrete voltages (V_1, V_2) at any time. In addition, if you were to view the signal over the entire bit period, changes in the signal voltage level occur only at the beginning and/or center point of the bit period. The voltage level and voltage transitions from one level to another represent the logical data value being transmitted.

PCM. For example, two binary PCM codes that are most frequently used in serial communications are Non Return-to-Zero-Level (NRZ-L) and Biphas-Level (Bi Φ -L).

NRZ-L. The simplest PCM format is NRZ-L. NRZ-L PCM specifies that when the transmitted signal has a voltage of V_1 , a logical "one" is being transmitted and when the signal has a voltage of V_2 , a logical "zero" is being transmitted. Figure 2 shows the relationship of the digital data being transmitted versus the corresponding PCM signal.

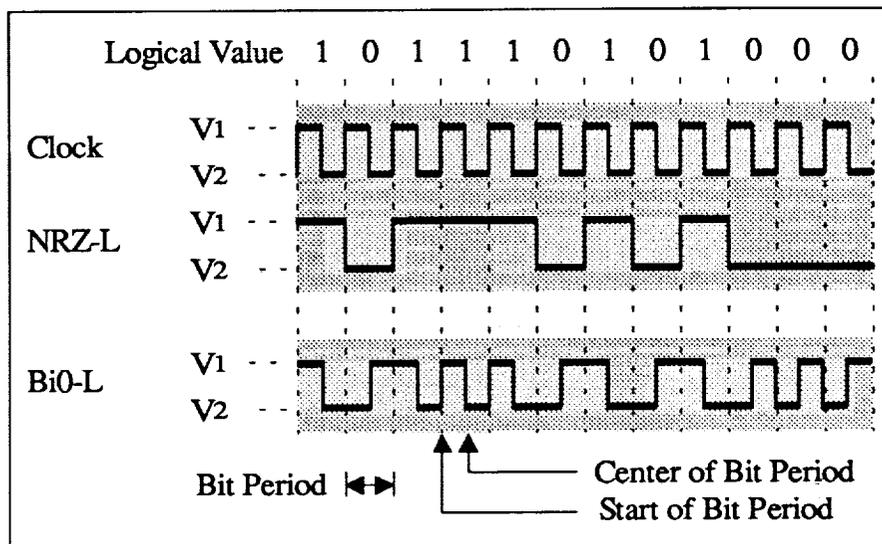


Figure 2. Examples of NRZ-L and Bi Φ -L PCM Codes

NRZ-L and Clock. One of the deficiencies of NRZ-L is that during transmission of a long sequence of all "zeros" or all "ones", the signal's transition density becomes low causing errors at the data receiver. These errors are due to timing variations between the data source and the data receiver. In this case, the data receiver is not able to identify the bit period boundaries due to the non-transitions within the transmitted signal, thus causing bit errors. To eliminate this problem in using NRZ-L, most digital equipment requires that a synchronous clock signal also be transmitted with the data signal. The clock and NRZ-L data signal relationship is also shown in Figure 2.

Bi Φ -L. Bi Φ -L is another popular PCM format because it eliminates the need for transmitting a clock signal with the data signal as in the case of NRZ-L. Bi Φ -L accomplishes this by combining the timing and data information into one signal. The bit period is divided into the start of the bit period and the center-bit period. A transition must always occur at the center-bit period location in the signal. These transitions are used

by the receiver equipment to extract or recover the synchronous clock from the PCM signal. The logical data value being transmitted is also represented during the center-bit period transition. A transition from V_1 to V_2 indicates a logical-one value and a transition from V_2 to V_1 indicates a logical-zero value.

PCM Code Conversion and Clock Regeneration

Much of the equipment used by the Ground Network (GN) accept only serial data in NRZ-L format with a synchronous clock signal provided; however, in some applications, the use of Bi Φ -L format has several advantages over NRZ-L. For example, since Bi Φ -L does not require an associated clock signal, the complexity of data switching systems is greatly reduced. In addition, Bi Φ -L eliminates the problem of data errors that can be caused by clock and data skew arising from the data and clock signal being transmitted across paths of different lengths.

In systems that use both NRZ-L with clock and Bi Φ -L, special interface devices are required to marry the PCM signals and receiver equipment (refer to Figure 3). These interface devices need to have the capability of accepting data in either NRZ-L or Bi Φ -L and providing a PCM code conversion to the alternative PCM format. This function is referred to as *PCM code conversion*. The devices must also be capable of extracting the clock information and producing a synchronous clock signal with the converted NRZ-L data for the receiver equipment. This function is referred to as *clock regeneration* or *clock recovery*. The CCCR and PCM bit synchronizer are two of these interface devices capable of both PCM code conversion and clock regeneration.

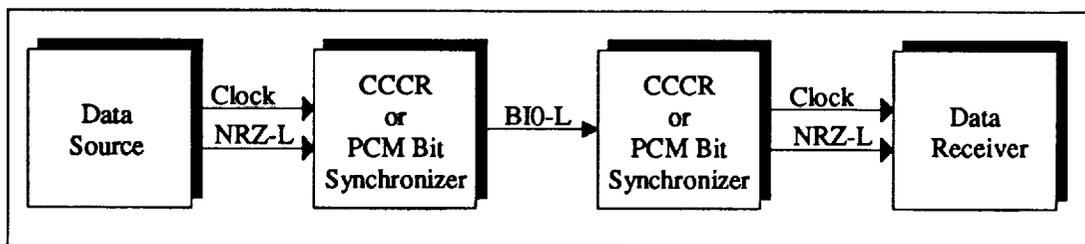


Figure 3. Example of PCM Code Conversion and Clock Regeneration

2. CCCR SYSTEM

General Description

The CCCR was designed and developed for the National Aeronautics and Space Administration (NASA) by AlliedSignal Technical Services Corporation (ATSC). The CCCR was designed and developed under the Bit Synchronizer Reduction (BSR) project for MIL. The BSR project and its applicability is detailed in Section 4. The CCCR consists of a 19-inch-wide by 10.5-inch-high by 23-inch-deep chassis housing two redundant power supplies and one to six CCCR Printed Circuit Boards (PCB). Local operation is performed through a front panel keyboard and fluorescent display. Figure 4 provides an illustration of the CCCR.

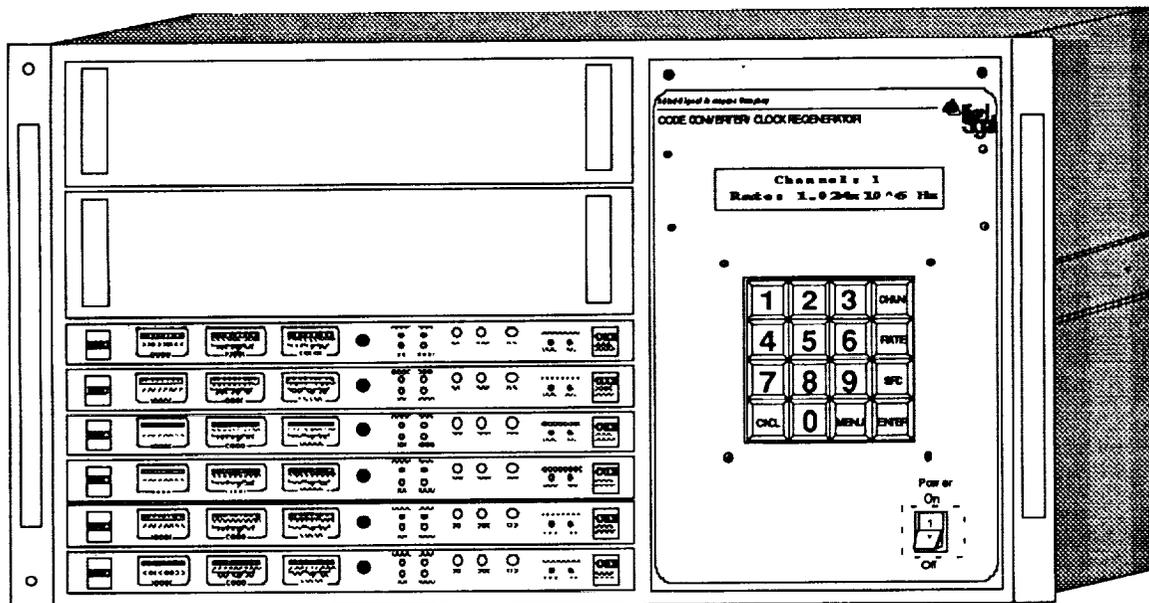


Figure 4. Code Converter/Clock Regenerator

Functional Capabilities

The following summarizes the capabilities of the CCCR:

Multiple-Channel. The CCCR system is configurable from one to six independent channels of PCM code conversion and clock regeneration.

Code Conversion and Clock Regeneration. Each CCCR channel is capable of accepting and performing PCM code conversion and clock regeneration on any of the following PCM codes: NRZ-L,M,S; Bi Φ -L,M,S; and Delay Modulation (DM)-M,S.

Source Input. Each CCCR channel is capable of accepting eight independent data sources and performing source selection.

Input Characteristics. The CCCR is configured to accept input signals with either Unipolar (TTL) or Bipolar (1 V_{pp} to 10 V_{pp}) voltage levels.

Dedicated NRZ-L and Clock Output. The CCCR provides a dedicated output signal pair consisting of the NRZ-L representation of the input data with a synchronous clock.

PCM Coded Output and Clock. A PCM coded output is provided for each CCCR channel and can be configured for any of the following PCM formats: NRZ-L,M,S, Bi Φ -L,M,S and DM-M,S. A synchronous clock associated with this output is also provided.

Fault Tolerance. Fault tolerance is provided by the use of redundant power supplies and in-line fuses on the power lines of each CCCR PCB.

Remote Operation. Remote configuration and monitoring is provided by a separate parallel data port for each CCCR channel.

VMEbus Compatibility. The CCCR PCBs have been designed such that they may also be used as a slave device in a VMEbus system.

Design Overview

The intent of the CCCR design is to provide the code conversion and clock regeneration capability at a low cost for environments in which signal conditioning is not required. Additionally, the intended application required the CCCR to provide a programmable data rate, thus the CCCR is capable of supporting data rates from 10 bps to 2.5 Mbps. Most low cost commercial code converter/clock recovery units can only be configured for one or several fixed rates.

Code Conversion and Clock Regeneration. The CCCR implements a simple algorithm for the code conversion and clock regeneration function. The algorithm only requires two Integrated Circuits (IC): a Numerically Controlled Oscillator (NCO) and a Electrically Programmable Logic Device (EPLD). A block diagram of this circuitry is shown in Figure 5.

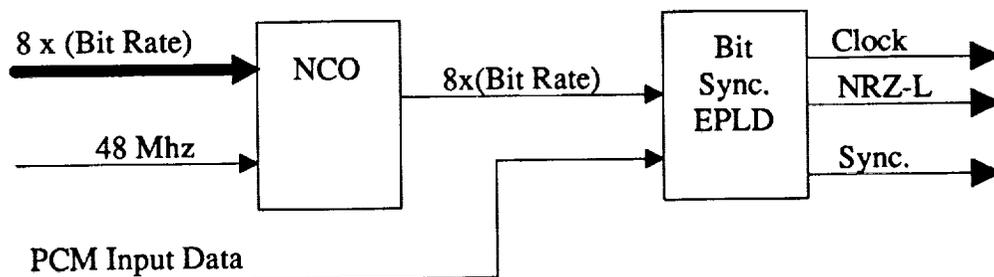


Figure 5. Code Conversion/Clock Regenerator Circuit Block Diagram

The NCO is loaded with a phase-value which is used with the input reference clock of 48 MHz to generate a precise frequency of 8 times the expected PCM bit rate. The 8 times clock is used by the EPLD to perform the actual code conversion and clock regeneration of the signal. The EPLD also produces a status signal indicating the synchronization (Sync.) status of the device to the incoming signal. A detailed block diagram of the Bit Sync. EPLD logic is shown in Figure 6.

For the eight PCM codes that are accepted by the CCCR (NRZ-L,M,S; BiΦ-L,M,S and DM-M,S), three pieces of information must be accumulated by the EPLD logic on the incoming PCM signal in order to identify the logical data values. Transitions in the signal need to be identified along with the direction of the transition (i.e. High-to-Low or Low-to-High). In addition, the Bit Synchronizer EPLD must correctly identify the beginning and center of the bit period based on the signal voltage transitions and the associated rules for the PCM format being accepted.

The algorithm used in the bit synchronizer EPLD is based on an eight times sampling of the signal over the bit period. Depending on the PCM code being received, the clock Generator is able to divide the eight times clock into a clock consistent with the bit period

of the data. The Clock Generator also synchronizes the clock to the incoming signal based on transitions in the data signal and the specified PCM characteristics.

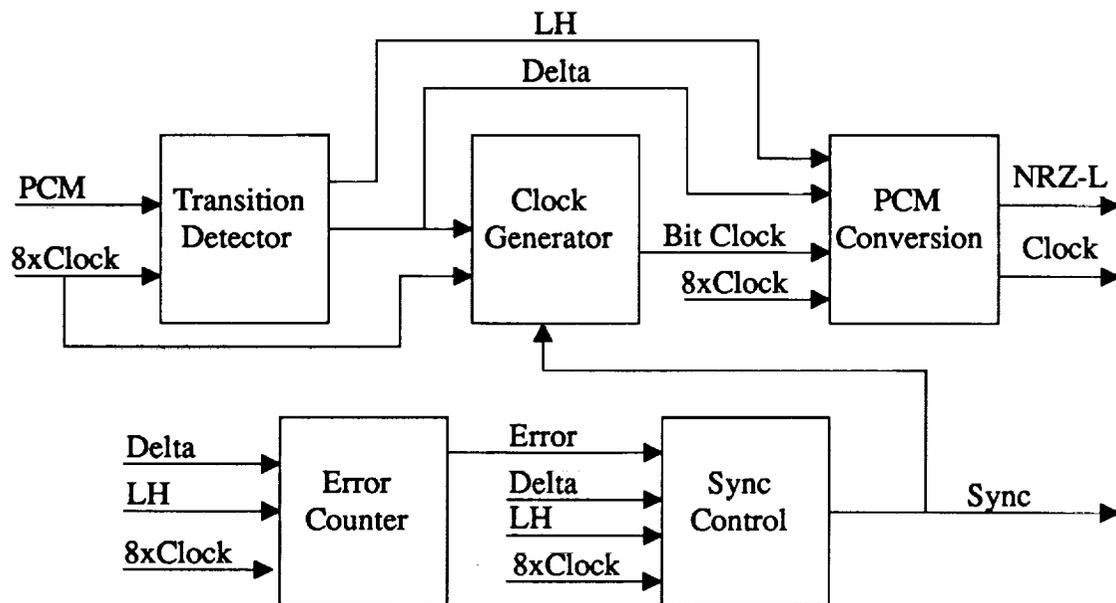


Figure 6. Functional Diagram of the Bit Synchronizer EPLD Logic

Using the generated clock, transition information and input PCM format the PCM Conversion functional block determines the logical data value associated with the input signal. Based on the rules for the selected PCM format being accepted, the Sync Control block indicates a lock on the incoming signal as long as the input PCM signal does not violate these rules. Each violation of the PCM rules, is registered by the Error Counter. If the Error Counter reaches its maximum value, an error signal is generated causing the Sync Control logic to indicate a loss of lock on the incoming PCM signal. Figure 7 provides a timing diagram for the internal bit synchronizer EPLD for an input in Bi Φ -L format.

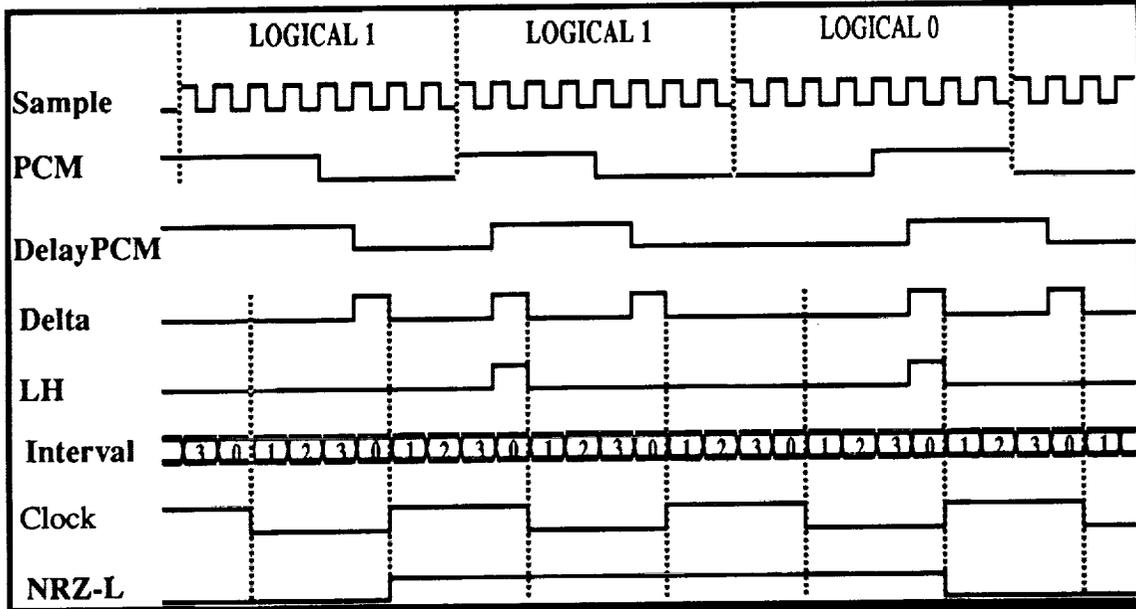


Figure 7. Bit Sync. EPLD Timing Diagram

This sampling method is a simple algorithm for determining the logical value of the input PCM signal; however, the output from the Bit Sync EPLD will contain jitter on the clock and NRZ-L data of up to one quarter of the bit period due to the sampling. To reduce the jitter on the output of the CCCR, the NRZ-L data is clocked into a data buffer prior to output. The data is then clocked out of the data buffer using a reduced rate clock from the NCO.

Since the clock used to shift data in and out of the buffer may vary slightly in frequency, a rate-adjust algorithm is implemented to prevent exceeding the storage capacities of the buffer, resulting in a loss of data. An embedded controller on the CCCR is used to calculate the approximate incoming data rate based on the clock generated in the Bit Sync EPLD. The following equation is used to determine this rate:

$$\text{Calculated Rate} = \text{Counter Value} / \text{Sample Time}$$

where $\text{Sample Time} = 1 / [(\text{Desired Precision}) * (\text{Configured Bit Rate})]$.

The NCO is then adjusted either slightly above or slightly below the calculated rate (taking into account the error of the calculation: $\text{Error} = 2/[\text{Sample Time}]$) based on a half-full signal generated by the data buffer.

In order to prevent excessive delays due to the buffering of data, a size programmable buffer was designed. The buffering is accomplished using two First-In-First-Out (FIFO) IC's and a buffer controller EPLD. A 64x8 bit FIFO is used for lower data rates. The buffer controller EPLD is capable of varying the data width into and out of the FIFO from 1 to 8 bits based on the input rate. For higher data rates, a 512x8 bit FIFO is used. Again the buffer controller is configured to operate at a word width from 1 to 8 bits. The embedded controller calculates the size of the buffering required and sets up the buffer controller based on the configured data rate. The buffer size required is calculated as follows:

$$\text{Buffer Size} = \text{Configured Bit Rate} / [(\text{FIFO Width}) / (\text{Maximum Allowable Delay})].$$

Figure 8 shows a diagram of the variable size buffer as implemented on the CCCR.

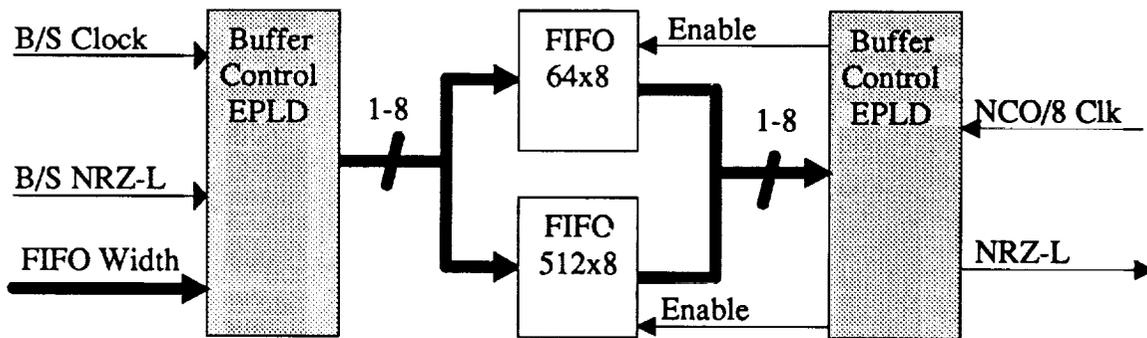


Figure 8. Variable Buffer Diagram

Finally, the NRZ-L data from the buffer is converted to one of the eight possible PCM formats by a code converter EPLD. The general rules used to generate the PCM data are described in Table 2. The entire design supporting all eight PCM codes is implemented in 12 ICs.

Table 2. PCM Format Definitions

PCM Format	Definition
NRZ-L	"One" is represented by V_1 . "Zero" is represented by V_2 .
NRZ-M	"One" is represented by a change in level. "Zero" is represented by no change in level.
NRZ-S	"One" is represented by no change in level. "Zero" is represented by a change in level.
Bi Φ -L	<i>Level change occurs at center of every bit period.</i> "One" is represented by a transition from V_1 to V_2 . "Zero" is represented by a transition from V_2 to V_1 .
Bi Φ -M	<i>Level change occurs at beginning of every bit period.</i> "One" is represented by a center bit period transition. "Zero" is represented by no center bit period transition.
Bi Φ -S	<i>Level change occurs at beginning of every bit period.</i> "One" is represented by no center bit period transition. "Zero" is represented by a center bit period transition.
DM-M	"One" is represented by a level change at the center bit period. "Zero" followed by a "zero" is represented by a level change at the end of the first "Zero" bit. No level change occurs when a "zero" is preceded by a "one".
DM-S	"Zero" is represented by a level change at the center bit period. "One" followed by a "one" is represented by a level change at the end of the first "one" bit. No level change occurs when a "one" is preceded by a "zero".

3. CCCR VERSUS PCM BIT SYNCHRONIZERS

Although the CCCR and PCM bit synchronizer perform the identical functions of PCM code conversion and clock regeneration, the devices are not interchangeable in all applications. The CCCR was designed specifically to provide a low-cost PCM code converter and clock regenerator unit that operates from 10 bps to 2.5 Mbps. The CCCR is also intended to be used in environments where noise considerations are negligible and signal voltage levels are deterministic.

PCM bit synchronizers have the additional capability of filtering, Automatic Gain Control (AGC), and direct current (dc) offset adjustments for noisy environments where signal characteristics may vary. In applications where cost is a major consideration and the added features of the PCM bit synchronizer are not required, the CCCR provides a low-cost alternative to the high-cost PCM bit synchronizer. Table 1 provides a comparison between features of the CCCR and PCM bit synchronizer.

Table 1. Summary of CCCR vs. PCM Bit Synchronizer Functions and Cost

Function	CCCR	PCM Bit Synchronizer
Filtering (Noise)	No	Yes
Input Amplitude	1 Vpp to 10 Vpp	.25 Vpp to 15 Vpp
Maximum dc Offset (based on a zero volt center)	40%	100%
Automatic Gain Control	No	Yes
PCM Code Conversion	Yes	Yes
Clock Recovery	Yes	Yes
Sync with External Clock	1 bit	< 1000 bits
Max. Input Sources	8	1-6
Remote Configuration	Yes	Yes
Bit Delay	Variable	Fixed
Per Channel Cost	\$2,400	\$5,000 - \$12,000

The following sections provide examples highlighting those applications best suited for the CCCR and those applications where the full features of a PCM bit synchronizer are required.

An Application Using PCM Bit Synchronizers

PCM bit synchronizers are extensively used at tracking stations and serve as interface devices between the RF equipment and data processing equipment. Figure 9 provides a simplified block diagram of this process. RF modulated spacecraft data is down-linked to the ground station where the receiver/demodulator RF equipment extract the serial PCM data stream. PCM bit synchronizers are then used to perform the PCM code conversion to NRZ-L format and the clock recovery function required for the data-processing equipment. It is common for the output from the receivers/demodulators to contain some undesirable signal characteristics such as noise, dc offset, varying amplitude and wave form rounding. Because of these signal characteristics, high performance PCM bit synchronizers with filtering, dc offset adjustment, AGC, and wave squaring are required for *signal conditioning*.

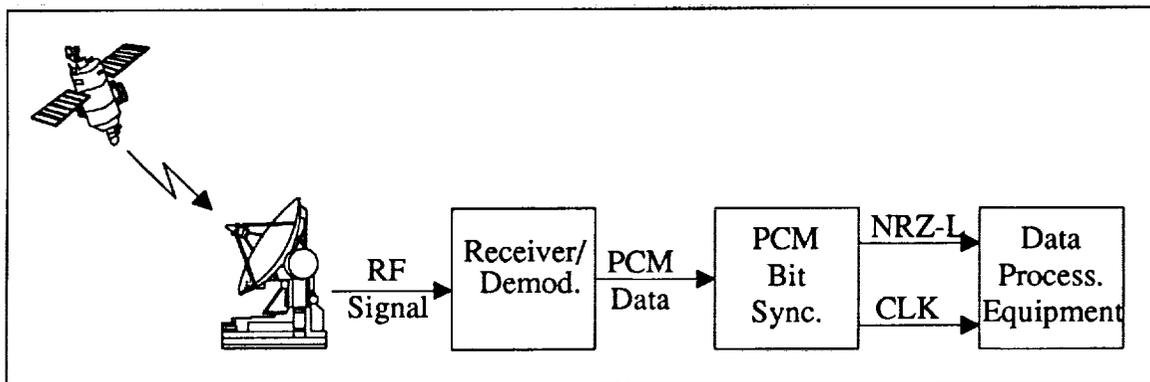


Figure 9. An Application of PCM Bit Synchronizers

An Application Using The CCCR

Figure 10 provides a detailed diagram of the tracking station example described in the previous section (Figure 9). The use of the high performance PCM bit synchronizer is the same as in the previous application; however, in this example, a switching system has been added along with a variety of data sources and data sinks. Bi Φ -L is used for routing the signal throughout the tracking station since the transfer of data only requires one signal for this code (refer to background section). The data-processing equipment require input in NRZ-L with a synchronous clock, so a single channel of the CCCR is provided for each input to the data equipment.

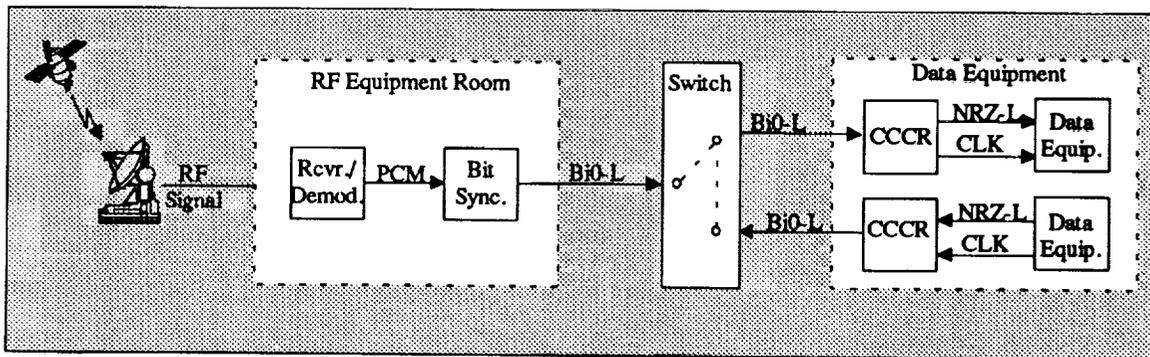


Figure 10. Application of the CCCR

Although high cost PCM bit synchronizers could be used for the code conversion and clock regeneration performed by the CCCR, in this example, this would have been expensive and unnecessary. Signal conditioning has been performed by the PCM bit synchronizers located in the RF equipment room and the distributed Bi Φ -L signal has a fixed voltage level (TTL or Bipolar) with minimal variations. This application is best suited for the lower cost CCCR units.

4. BIT SYNCHRONIZER REDUCTION PROJECT

The CCCR was designed and developed under the Bit Synchronizer Reduction (BSR) project for the Merritt Island Tracking Station (MIL). A total of 48 PCM bit synchronizers are used at MIL, configured as shown in Figure 11.

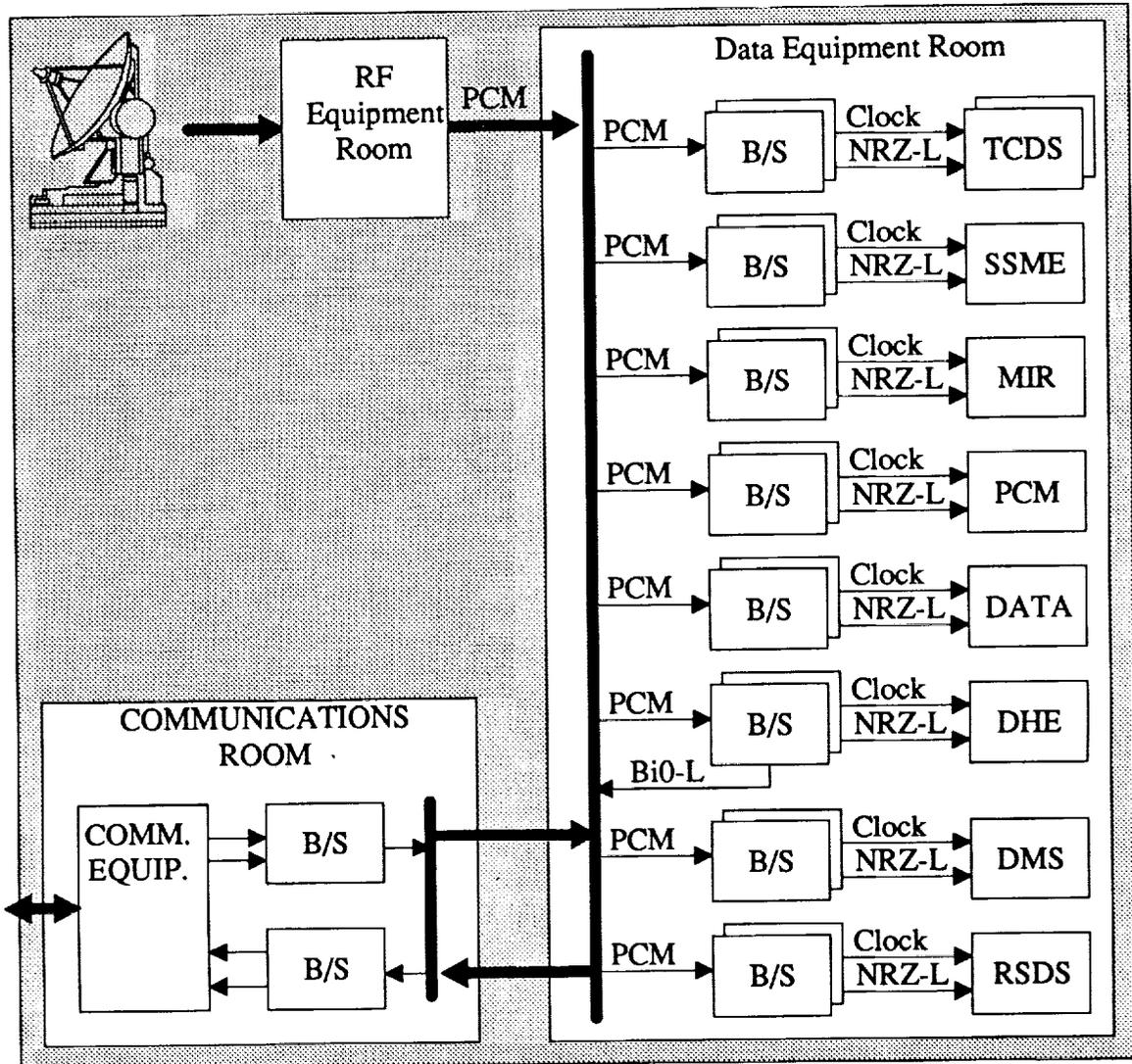


Figure 11. MIL PCM Bit Synchronizer Configuration Before (BSR)

In this application, the high performance PCM bit synchronizers are not located within the RF system, but are distributed within the data processing system. Every input to each of the data systems shown uses a high-performance PCM bit synchronizer for signal conditioning and PCM code conversion of the signals from the RF equipment room. Two models of PCM bit synchronizers are used in this system: an earlier series with lower performance (37 units) and a newer digital model with higher performance capability (11 units).

The increasing age and obsolescence of the older model PCM bit synchronizer is resulting in an increased cost in sustaining these units. A considerable cost savings would be achieved by reducing the maintenance required for the PCM bit synchronizers. The most direct method of reducing the maintenance cost is to replace all the older PCM bit synchronizers with newer, low maintenance PCM bit synchronizers. This approach was determined to be too costly based on current commercial prices.

Instead of replacement, an alternate approach to reduce the number of PCM bit synchronizers was selected for MIL. In this approach, the newer model PCM bit synchronizer will be used to accept and perform signal conditioning on the signals from the RF Equipment Room. The output from these high performance PCM bit synchronizers would then be distributed to the other data equipment. The many channels of code conversion required for the data equipment will to be performed by lower cost CCCR systems developed specifically for this application. Figure 12 shows the MIL configuration using the CCCR systems.

Using the CCCR units, all 37 of the older PCM bit synchronizers will be eliminated. The remaining 11 higher performance PCM bit synchronizers will be combined with 7 similar spare units to provide the initial processing of the PCM data originating from the RF Equipment Room (refer to Figure 12). Nine CCCR units will be delivered for use as PCM code converters and clock regenerators for the remaining data processing equipment.

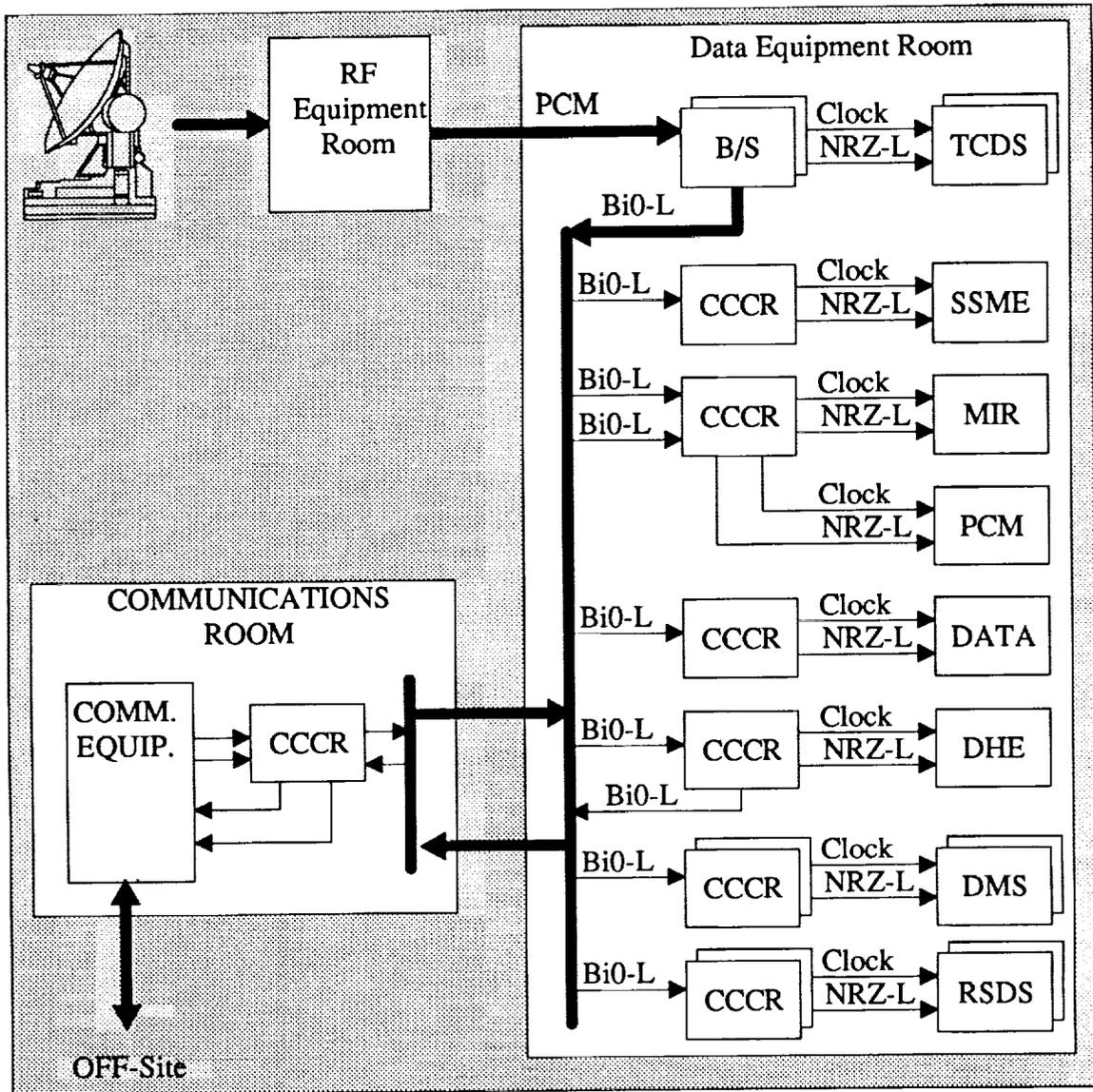


Figure 12. MIL Configuration Using CCCR Systems

5. SUMMARY

The low cost benefits of the CCCR are exemplified by the application of the units at MIL. It is estimated that approximately \$56,000 a year is being expended in maintenance at MIL on the older PCM bit synchronizers. By implementing the reduction using CCCR units as described in Section 4, a cost savings of \$150,000 to \$350,000 is achieved over estimated replacement cost of \$250,000 to \$450,000. In addition, since the CCCR requires essentially no adjustments, the maintenance cost for MIL is reduced by approximately \$56,000 per year. The CCCR units are expected to be used until the MIL system is completely redesigned in approximately four years, which results in a total savings from maintenance of \$224,000.

Another benefit resulting from the implementation of the CCCR units is that an overall performance increase in the system was obtained. The maximum bit rate supported by the older PCM bit synchronizers was 1.2 Mbps. With the use of the CCCR units at MIL, the overall maximum bit rate is 2.5 Mbps, an increase of over 50 percent.

6. CONCLUSION

The CCCR is a low-cost PCM Code Converter/Clock Regenerator that has been specifically designed for communications systems not requiring signal conditioning. Often, high-cost PCM bit synchronizers are used in these applications since commercial code converters and clock recovery units operate only at fixed frequencies. The CCCR provides an alternative to the PCM bit synchronizers by performing code conversion and clock regeneration at rates from 10 bps to 2.5 Mbps.



ARRAY SIMULATIONS PLATFORM (ASP)
PREDICTS
NASA DATA LINK MODULE (NDLM)
PERFORMANCE¹

Allen David Snook
Loral Aerosys SEAS

Through a variety of imbedded theoretical and actual antenna patterns, the array simulation platform (ASP) enhanced analysis of the array antenna pattern effects for the KTx (Ku-Band Transmit) service of the NDLM (NASA Data Link Module). The ASP utilizes internally stored models of the NDLM antennas and can develop the overall pattern of antenna arrays through common array calculation techniques.

ASP expertly assisted in the diagnosing of element phase shifter errors during KTx testing, and was able to accurately predict the overall array pattern from combinations of the four internally held element patterns. This paper provides an overview of the use of the ASP software in the solving of array mis-phasing problems.

I. Introduction

One path of development for highly directional antenna assemblies is the use of a matrix, or array, or antenna elements arranged in such a manner as to produce a highly focused beam of transmit energy, or to serve as a highly selective receive antenna. The use of antenna arrays to perform these functions dates back to the 1920s.²

Similarly, the use of the modern computer and specialty software to calculate patterns for arrays of antenna elements is not a new science, either. The ASP differs from previous work not on the merits of it's code, but the application of the code towards the solving of array alignment problems.

¹ Developed for NDLM testing in support of R. Stelmaszek, GSFC Code 531.4, Networks Test Section, and K. Perko, GSFC Code 737.2, Instrument and Advanced Development Section.

² W. L. Stutzman and G. A. Thiele, "Antenna Theory and Design," Wiley, 1981, Ch. 3

II. Ideal Application

The NASA Data Link Module (NDLM) array antenna, pictured in Figure 1, is comprised of several individual communications antenna experiments. Of these, the KTx (Ku-Band Transmit) portion of the NDLM is of unique interest. The KTx is the only subsystem on the NDLM package that possesses a beam-steering capability. Through the use of mechanical phase shifters, the four elements comprising the KTx subsystem can be moved in phase relative to each other. The elements of the KTx are configured in an arrangement known as a **planar array**, with a rectangular disposition.

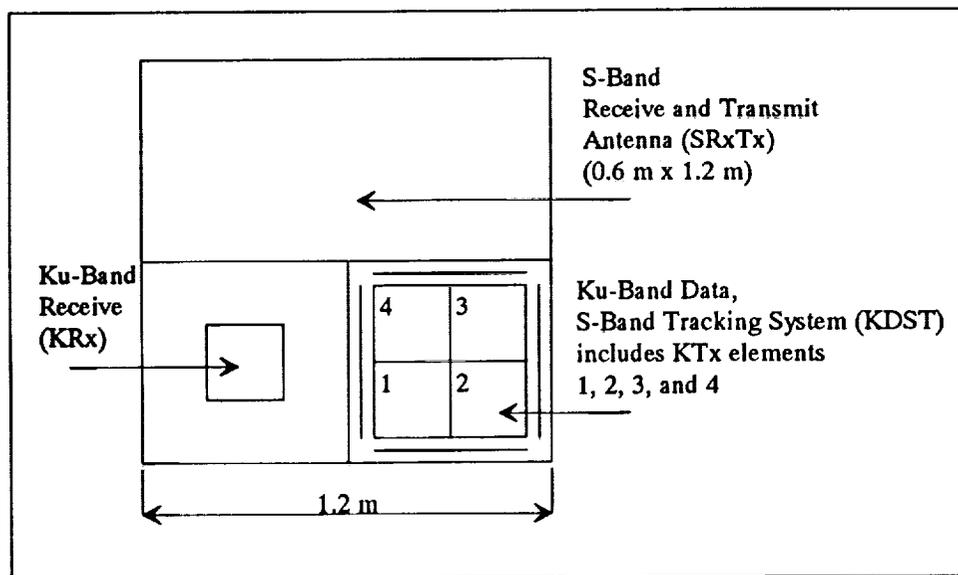


Figure 1. NDLM illustrated

The solitary data structure upon which the ASP software acts is, of course, the antenna element specifications. The variables of interest for each antenna element are 1) location in x, y, z space, 2) the relative amplitude, 3) the relative phase, and 4) the element type. As illustrated in Figure 2, the KTx element centers, form a square in the xz plane, with the basis of the square as 0.279 m. (Element spacing, at the principal KTx operating frequency of 15.0034 GHz, is about 14λ).³ As readers with experience in array construction are aware, that large a spacing will immediately contribute to the creation of numerous additional lobes in the overall array pattern.

³

λ , wavelengths

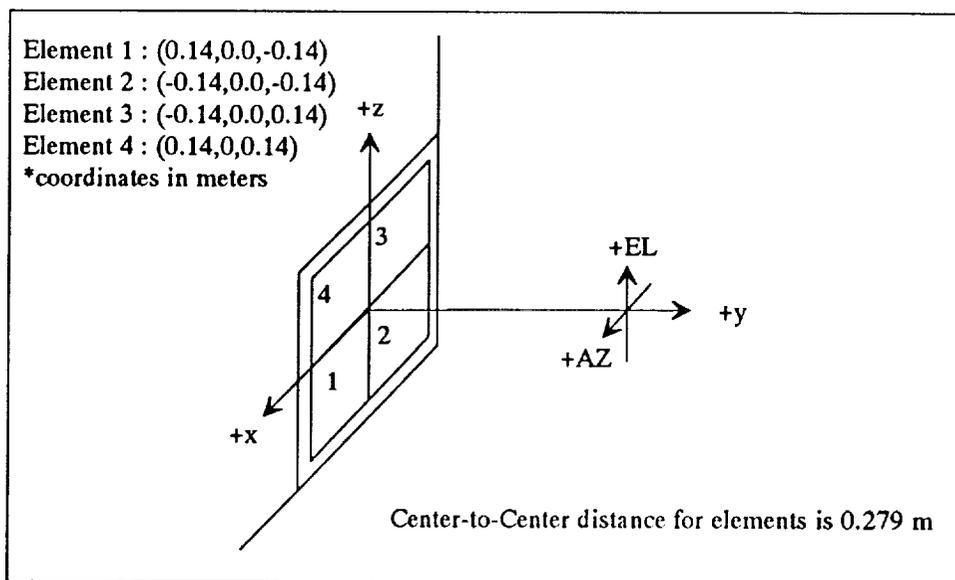


Figure 2. KTx geometry

It is not the purpose of this paper to provide a treatise on array antenna dynamics, as much work has already been covered on the subject. For a comprehensive discussion of the subject, the reader is referred to the previously referenced text by Stutzman (1981). For the purposes of this paper, a modest understanding of array factors is recommended.

The ASP was asked to prepare an analysis of a planar array of four isotropic sources, $F(\theta, \phi) = 1.0$, with an inter-element basis of 27.9 cm, an operating frequency of 15.0034 GHz, identical relative amplitudes and phases. This exercise performs, in essence, an analysis of the KTx, with the effects of the element patterns removed. The array pattern is given in Figure 3.

The previously mentioned grating lobes would appear to ruin the chances of any directivity ever being obtained from an antenna with this geometry, but bear in mind the removal of the effect of the KTx element pattern. KTx element patterns were taken on several occasions, and the average pattern data for each element in azimuth and elevation was collected and integrated into the code for ASP. Off-axis data was estimated using an interpolation function, as no off-axis data was readily available for the elements. The patterns were discretized in half-degree increments, with linear interpolation providing the pattern values for the remainder of the curves. The patterns for each element are presented in Figure 4a through d for elements 1 through 4, respectively.

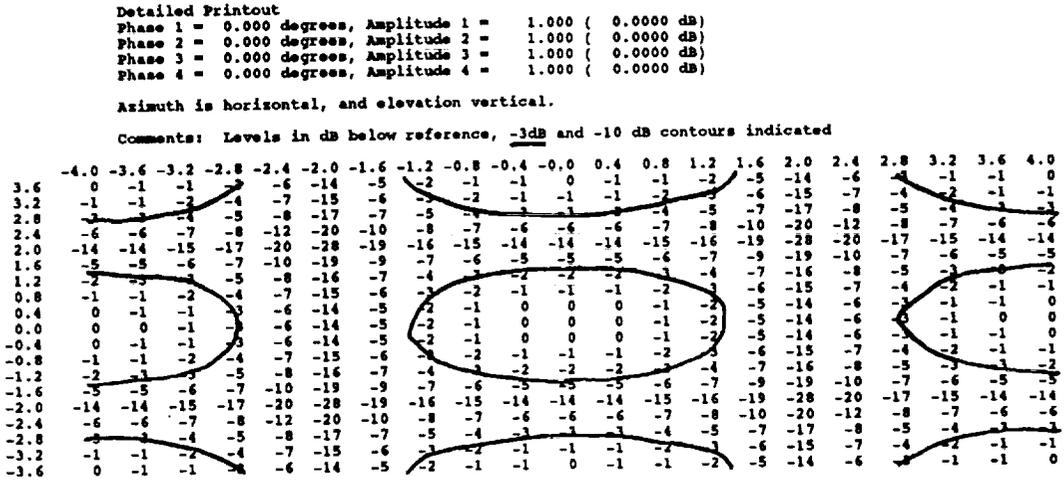


Figure 3. Pattern for an Array of Isotropic Elements, KTx Basis

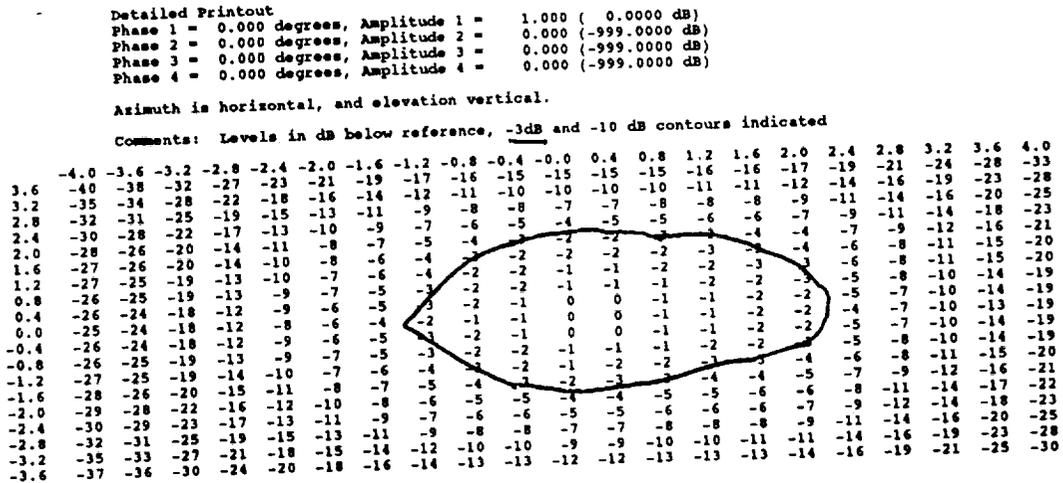


Figure 4a. Element 1 Amplitude Pattern

Detailed Printout

Phase 1 = 0.000 degrees, Amplitude 1 = 0.000 (-999.0000 dB)
Phase 2 = 0.000 degrees, Amplitude 2 = 1.000 (0.0000 dB)
Phase 3 = 0.000 degrees, Amplitude 3 = 0.000 (-999.0000 dB)
Phase 4 = 0.000 degrees, Amplitude 4 = 0.000 (-999.0000 dB)

Asimuth is horizontal, and elevation vertical.

Comments: Levels in dB below reference, -3dB and -10 dB contours indicated

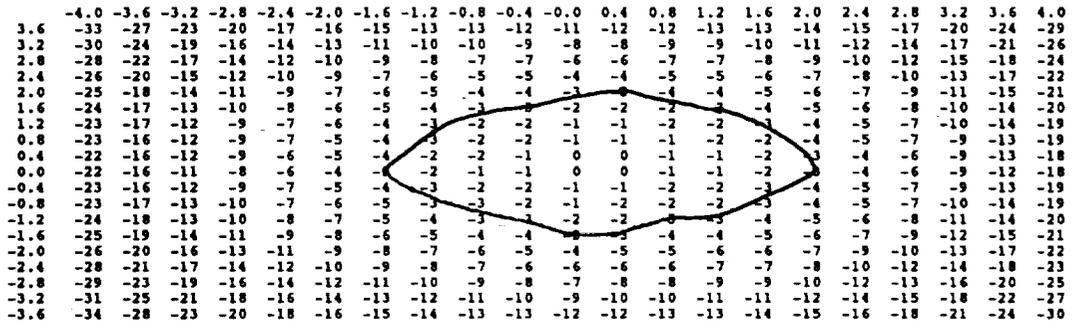


Figure 4b. Element 2 Amplitude Pattern

Detailed Printout

Phase 1 = 0.000 degrees, Amplitude 1 = 0.000 (-999.0000 dB)
Phase 2 = 0.000 degrees, Amplitude 2 = 0.000 (-999.0000 dB)
Phase 3 = 0.000 degrees, Amplitude 3 = 1.000 (0.0000 dB)
Phase 4 = 0.000 degrees, Amplitude 4 = 0.000 (-999.0000 dB)

Asimuth is horizontal, and elevation vertical.

Comments: Levels in dB below reference, -3dB and -10 dB contours indicated

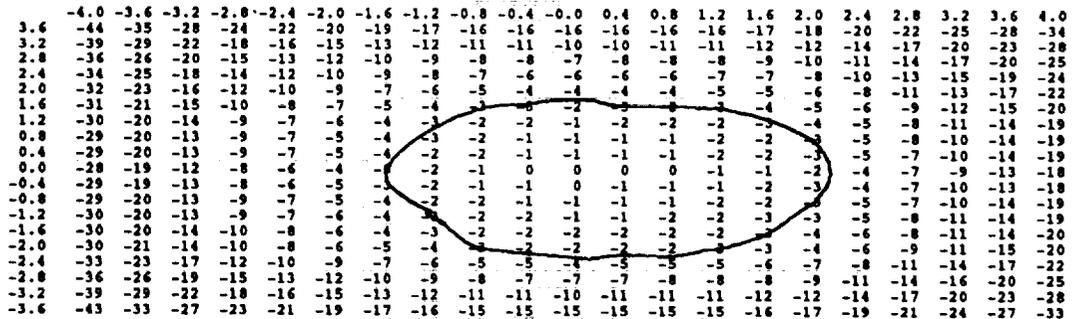


Figure 4c. Element 3 Amplitude Pattern

Detailed Printout
Phase 1 = 0.000 degrees, Amplitude 1 = 0.000 (-999.0000 dB)
Phase 2 = 0.000 degrees, Amplitude 2 = 0.000 (-999.0000 dB)
Phase 3 = 0.000 degrees, Amplitude 3 = 0.000 (-999.0000 dB)
Phase 4 = 0.000 degrees, Amplitude 4 = 1.000 (0.0000 dB)

Azimuth is horizontal, and elevation vertical.

Comments: Levels in dB below reference, -3dB and -10 dB contours indicated

	-4.0	-3.6	-3.2	-2.8	-2.4	-2.0	-1.6	-1.2	-0.8	-0.4	0.0	0.4	0.8	1.2	1.6	2.0	2.4	2.8	3.2	3.6	4.0
3.6	-34	-28	-25	-22	-20	-18	-17	-16	-16	-15	-15	-16	-16	-17	-18	-19	-22	-25	-28	-34	
3.2	-29	-23	-20	-17	-15	-14	-12	-11	-11	-10	-10	-11	-11	-11	-12	-13	-14	-17	-20	-23	-30
2.8	-26	-20	-16	-14	-12	-10	-9	-8	-7	-7	-7	-7	-8	-8	-10	-11	-14	-16	-20	-26	
2.4	-23	-18	-14	-11	-9	-8	-6	-6	-5	-5	-4	-4	-5	-5	-6	-7	-8	-11	-14	-17	-24
2.0	-22	-16	-13	-10	-8	-6	-5	-4	-4	-3	-3	-3	-4	-4	-5	-6	-7	-10	-13	-16	-22
1.6	-21	-15	-12	-9	-7	-5	-4	-3	-2	-2	-2	-2	-2	-3	-4	-5	-6	-9	-11	-15	-21
1.2	-20	-15	-11	-8	-6	-5	-3	-2	-2	-1	-1	-1	-1	-2	-3	-4	-5	-8	-10	-14	-20
0.8	-20	-14	-11	-8	-6	-4	-3	-2	-1	0	0	0	-1	-1	-2	-3	-4	-7	-10	-13	-20
0.4	-19	-14	-10	-8	-5	-4	-3	-2	-1	0	0	0	-1	-1	-2	-3	-4	-7	-10	-13	-20
0.0	-19	-13	-10	-7	-5	-4	-3	-2	-1	0	0	-1	-1	-2	-2	-3	-4	-7	-10	-13	-20
-0.4	-19	-14	-10	-8	-5	-4	-3	-2	-2	-1	-1	-1	-2	-2	-3	-4	-5	-8	-11	-14	-20
-0.8	-20	-14	-11	-8	-6	-4	-3	-2	-2	-1	-1	-2	-2	-3	-3	-4	-6	-8	-11	-14	-21
-1.2	-20	-15	-11	-9	-6	-5	-4	-3	-2	-2	-2	-3	-3	-4	-5	-6	-9	-12	-15	-22	
-1.6	-21	-16	-12	-9	-7	-5	-4	-3	-2	-2	-3	-3	-4	-4	-5	-6	-7	-10	-13	-16	-22
-2.0	-22	-16	-13	-10	-8	-6	-5	-4	-3	-3	-4	-4	-5	-5	-6	-7	-8	-9	-12	-15	-24
-2.4	-24	-18	-15	-12	-10	-8	-7	-6	-6	-5	-5	-6	-6	-7	-8	-9	-10	-11	-14	-17	-27
-2.8	-26	-21	-17	-15	-12	-11	-10	-9	-8	-8	-7	-8	-8	-9	-10	-11	-12	-13	-14	-17	-29
-3.2	-29	-23	-20	-17	-15	-13	-12	-11	-11	-10	-10	-10	-11	-11	-12	-13	-14	-17	-20	-23	-29
-3.6	-32	-27	-23	-20	-18	-17	-15	-15	-14	-14	-13	-13	-14	-14	-15	-16	-17	-20	-23	-26	-33

Figure 4d. Element 4 Amplitude Pattern

The ability of the software to model the elements and their environment was limited by the amount of data available upon which to construct the virtual antenna. For this reason, the ASP was unable to make use of element *phase vs. angle* data--only amplitude vs. angle data was available. It would be advantageous, for the sake of competition in the simulated environment, to obtain and implement this data in future uses of the ASP software.

Again, returning to the linear array of two isotropic elements, the replacement of the isotropic elements with the appropriate synthesized KTx patterns results in the following result for an analysis of A1 and A4, again with equal amplitude and phase. In this representation, the A14 elements will produce a pattern affected by array factors only in the elevation pattern. This is a significant improvement in directivity over the isotropic example of Figure 4.

But wait! I thought this KTx was a two dimensional array! Why, yes, it is. Expanding our example to include all four elements, with their individual patterns, identical relative amplitudes and phases, results in the pattern in Figure 6 obtained from ASP at 15.0034 GHz.

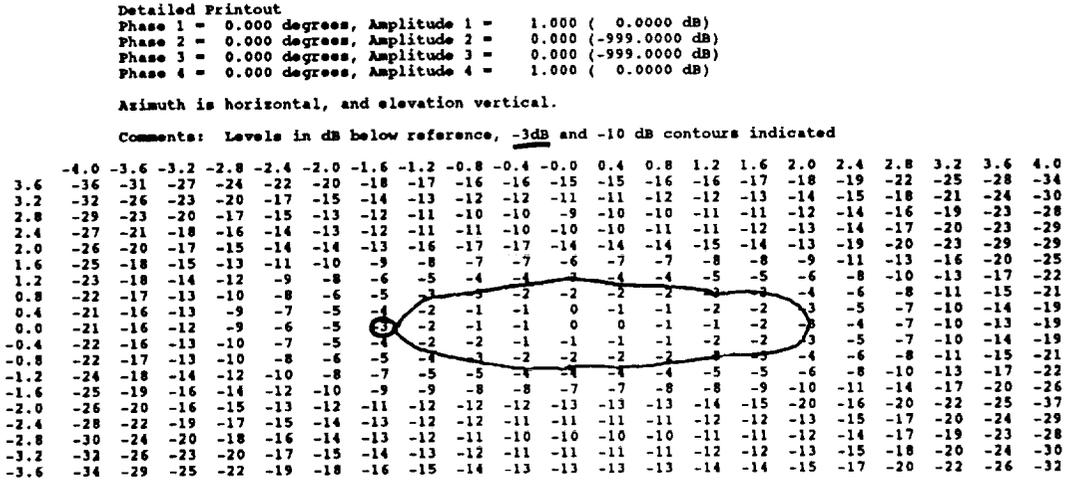


Figure 5. Pattern for 1 & 4 Synthesized Elements, KTx Basis

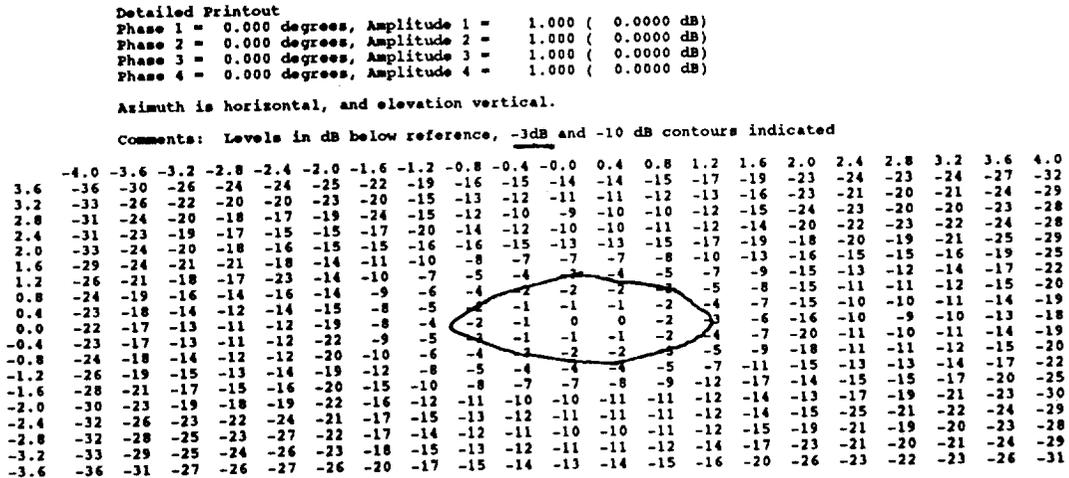


Figure 6. Complete pattern for all elements active

The overall gain the pattern of Figure 6 should exceed an individual elements gain by 6 dB, due to the summation of the gains (0 dB) of each of the four elements. Of course, if the real world produced series of individual antennas with precisely identical gains, and precisely phased networks to drive those antennas, the work would stop at this point. The true is far stranger than that fiction. For the KTx antennas, the following relative amplitudes were observed for the individual antenna beam peaks (Table 1.) The relative gain measurements were obtained from in-field measurements, and were used throughout KTx ASP calculations.

Table 1. Relative KTx element main-beam gains

Element	Relative Gain (dB)	Decimalized Gain
1	-1.3	0.7413
2	-1.8	0.6607
3	-2.7	0.5370
4	+0.0	1.0000

Further expanding to include these differential gains, and with identical relative phases, results in the following patterns obtained from ASP at 15.0034 GHz:

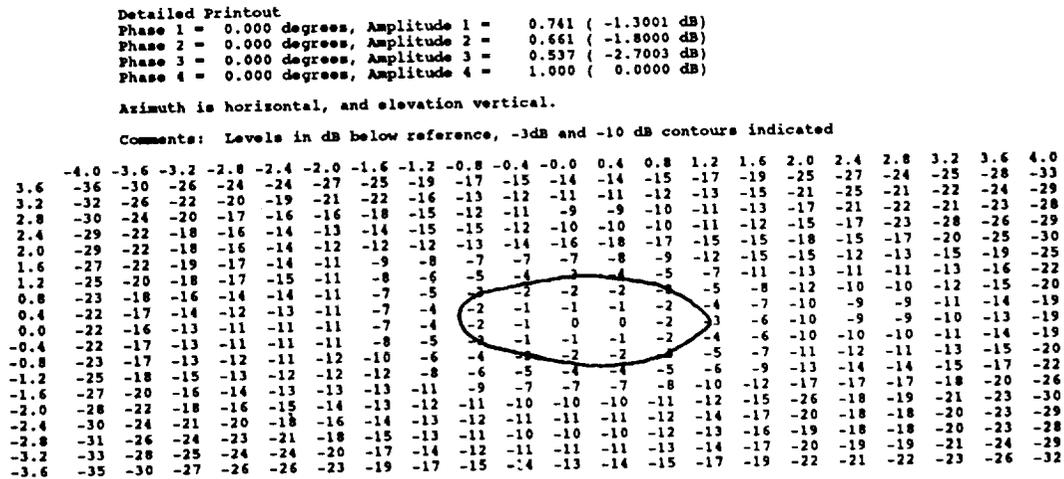


Figure 7. Complete pattern for all elements, exact amplitudes

The overall main-beam gain drops to about 4.6 dB, due to the summation of the less than ideal main-beam gains of each element. This pattern is the best pattern to be expected of the KTx subsystem, provided all elements are co-phased.

III. Non-ideal Phase Application

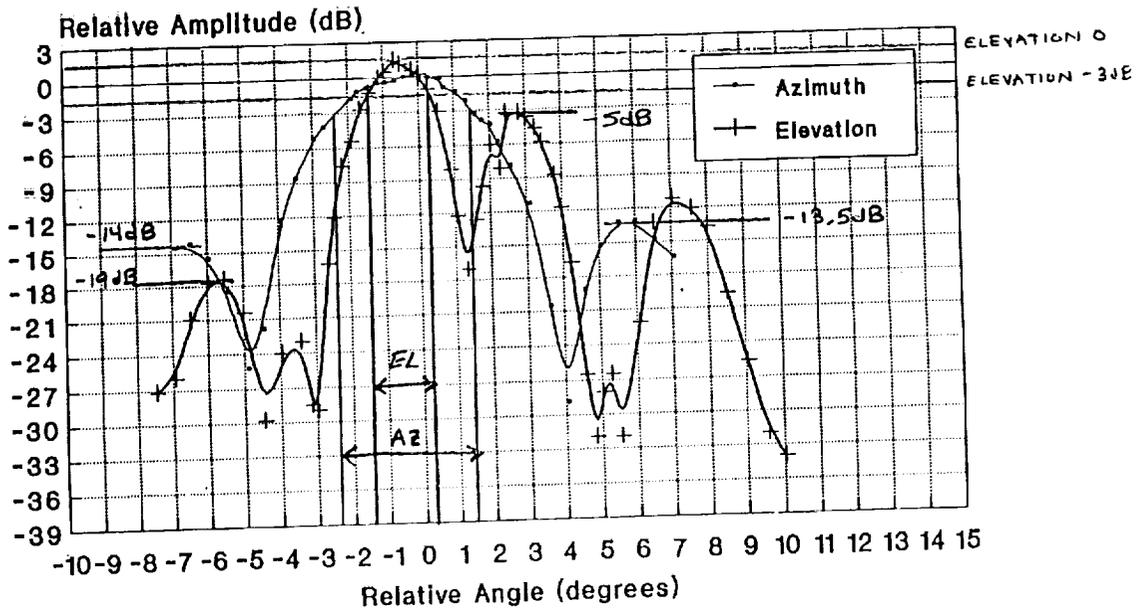
The pattern presented in Figure 8 was obtained for the KTx with elements 1 & 4 active. It is from this pattern that an off-axis skew of the main beam, and some assymetry in side lobe levels is apparent. The variable that could be manipulated to correct this assymetry is the relative phase between the elements. For example, ASP was asked to show the azimuthal pattern when elements when elements 1 and 4 were cophased 60° behind the 2 and 3 pair. This should result in a beam skewed in azimuth, favoring the 1 and 4 pair. This is precisely what ASP generated, the results of which are indicated in Figure 9.

The next step was to run ASP through several simulations, with elemental phases being modified in a set pattern, to derive the overall relation between misphasing of the elements, and mispointing of the main beam. After several iterations, the following was determined: For every 30° in electrical phase between two elements, the beam would move along a line between the elements 0.15° in spherical coordinates. Additionally, changes in phase between elements had a dramatic effect on the effect of the array factor on those elements, discouraging and encouraging the sidelobe levels of the overall pattern.

KTx Az & El Patterns

A14 SUBARRAY

RFSOC 8-28-92



File A14828

DS5 A14

3 dB BW	AZ	3.8°	
	EL	1.8°	
SLL	AZ	-14 dB	-13.5 dB
	EL	-19 dB	-5 dB

Figure TBD
DS5

Figure 8. Actual KTx 1 & 4 Combined Pattern⁴

⁴ Taken from SSG-92-205, pg. 50, Pattern DS-5, Aug 28, 1992

Detailed Printout
Phase 1 = 300.000 degrees, Amplitude 1 = 0.741 (-1.3001 dB)
Phase 2 = 0.000 degrees, Amplitude 2 = 0.661 (-1.8000 dB)
Phase 3 = 0.000 degrees, Amplitude 3 = 0.537 (-2.7003 dB)
Phase 4 = 300.000 degrees, Amplitude 4 = 1.000 (0.0000 dB)

Azimuth is horizontal, and elevation vertical.

Comments: Levels in dB below reference, -3dB and -10 dB contours indicated

	-4.0	-3.6	-3.2	-2.8	-2.4	-2.0	-1.6	-1.2	-0.8	-0.4	-0.0	0.4	0.8	1.2	1.6	2.0	2.4	2.8	3.2	3.6	4.0	
3.6	-37	-30	-25	-22	-21	-20	-22	-30	-20	-17	-15	-14	-14	-15	-16	-18	-22	-28	-29	-30	-33	
3.2	-33	-27	-22	-19	-17	-16	-17	-21	-17	-14	-11	-11	-11	-11	-13	-15	-18	-25	-28	-27	-30	
2.8	-32	-25	-20	-17	-15	-14	-14	-16	-16	-13	-10	-10	-10	-10	-11	-13	-15	-20	-32	-27	-29	
2.4	-30	-24	-19	-16	-13	-12	-11	-12	-14	-16	-13	-11	-10	-10	-11	-12	-14	-17	-22	-30	-31	
2.0	-28	-22	-19	-17	-15	-13	-11	-10	-8	-7	-6	-6	-7	-8	-10	-13	-16	-15	-16	-19	-25	
1.6	-25	-20	-17	-15	-13	-12	-12	-9	-6	-5	-3	-3	-4	-5	-6	-9	-14	-15	-15	-17	-22	
1.2	-24	-18	-15	-13	-12	-12	-11	-8	-5	-3	-2	-2	-2	-3	-4	-7	-12	-15	-14	-16	-20	
0.8	-23	-17	-13	-11	-9	-8	-9	-8	-4	-2	-1	0	-1	-2	-3	-5	-10	-14	-13	-15	-19	
0.4	-22	-16	-12	-9	-7	-7	-9	-8	-4	-2	0	0	0	-1	-2	-5	-9	-15	-13	-15	-19	
0.0	-22	-16	-11	-9	-8	-8	-9	-8	-4	-2	-1	-1	-1	-2	-3	-5	-9	-15	-14	-16	-20	
-0.4	-23	-16	-12	-9	-7	-7	-9	-8	-4	-2	0	0	0	-1	-2	-5	-9	-15	-13	-15	-19	
-0.8	-24	-17	-13	-10	-8	-8	-9	-8	-4	-2	-1	-1	-1	-2	-3	-5	-9	-15	-14	-16	-20	
-1.2	-26	-19	-13	-11	-8	-8	-9	-10	-6	-4	-2	-2	-2	-3	-4	-6	-10	-16	-16	-17	-21	
-1.6	-27	-21	-14	-11	-10	-9	-10	-11	-6	-4	-2	-2	-2	-3	-4	-6	-8	-11	-17	-19	-23	
-2.0	-28	-21	-15	-13	-12	-11	-11	-13	-11	-9	-7	-7	-7	-8	-9	-12	-14	-19	-23	-23	-26	
-2.4	-29	-23	-16	-14	-13	-13	-13	-14	-13	-11	-10	-10	-10	-10	-12	-15	-17	-20	-22	-24	-30	
-2.8	-30	-24	-17	-15	-14	-14	-14	-15	-13	-11	-10	-10	-10	-11	-12	-13	-15	-20	-21	-21	-24	-30
-3.2	-32	-26	-18	-17	-17	-17	-17	-18	-16	-13	-11	-10	-10	-10	-11	-13	-15	-20	-22	-22	-24	-29
-3.6	-34	-29	-20	-19	-19	-19	-20	-20	-17	-14	-12	-11	-11	-11	-12	-14	-16	-21	-24	-23	-25	-30

Figure 9. Skew demonstration

Figures 10a through 10g are included as a demonstration of this beam steering effect, in which one pair (1,4) of adjacent elements are moved in 30° increments in phase away from the other co-phased pair (2,3). Notice the increase in side lobe levels as the differential becomes larger.

Detailed Printout

Phase 1 = 0.000 degrees, Amplitude 1 = 0.741 (-1.3001 dB)
Phase 2 = 0.000 degrees, Amplitude 2 = 0.661 (-1.8000 dB)
Phase 3 = 0.000 degrees, Amplitude 3 = 0.537 (-2.7003 dB)
Phase 4 = 0.000 degrees, Amplitude 4 = 1.000 (0.0000 dB)

Azimuth is horizontal, and elevation vertical.

Comments: Levels in dB below reference, -3dB and -10 dB contours indicated

	-4.0	-3.6	-3.2	-2.8	-2.4	-2.0	-1.6	-1.2	-0.8	-0.4	0.0	0.4	0.8	1.2	1.6	2.0	2.4	2.8	3.2	3.6	4.0
3.6	-36	-30	-26	-24	-24	-27	-25	-19	-17	-15	-14	-14	-15	-17	-19	-25	-27	-24	-25	-28	-33
3.2	-32	-26	-22	-20	-19	-21	-22	-16	-13	-12	-11	-11	-12	-13	-15	-21	-25	-21	-22	-24	-29
2.8	-30	-24	-20	-17	-16	-16	-18	-15	-12	-11	-9	-9	-10	-11	-13	-17	-21	-22	-21	-23	-28
2.4	-29	-22	-18	-16	-14	-13	-14	-15	-12	-10	-10	-10	-11	-12	-15	-17	-23	-28	-26	-29	
2.0	-29	-22	-18	-16	-14	-12	-12	-13	-14	-16	-18	-17	-15	-15	-18	-15	-17	-20	-25	-30	
1.6	-27	-22	-19	-17	-14	-11	-9	-8	-7	-7	-8	-9	-12	-15	-15	-12	-13	-15	-19	-25	
1.2	-25	-20	-18	-17	-15	-11	-8	-6	-5	-4	-4	-5	-7	-11	-13	-11	-11	-13	-16	-22	
0.8	-23	-18	-16	-14	-14	-11	-7	-5	-3	-2	-2	-3	-5	-8	-12	-10	-10	-12	-15	-20	
0.4	-22	-17	-14	-12	-13	-11	-7	-4	-2	-1	-1	-1	-2	-4	-7	-10	-9	-9	-11	-14	-19
0.0	-22	-16	-13	-11	-11	-11	-7	-4	-2	-1	0	0	-2	-3	-6	-10	-9	-9	-10	-13	-19
-0.4	-22	-17	-13	-11	-11	-11	-8	-5	-1	-1	-1	-1	-2	-4	-6	-10	-10	-10	-11	-14	-19
-0.8	-23	-17	-13	-12	-11	-12	-10	-6	-4	-2	-2	-3	-5	-7	-11	-12	-11	-13	-15	-20	
-1.2	-25	-18	-15	-13	-12	-12	-12	-8	-6	-5	-4	-4	-5	-6	-9	-13	-14	-14	-15	-17	-22
-1.6	-27	-20	-16	-14	-13	-13	-11	-9	-7	-7	-7	-8	-10	-12	-17	-17	-17	-18	-20	-26	
-2.0	-28	-22	-18	-16	-15	-14	-13	-12	-11	-10	-10	-10	-11	-12	-15	-26	-18	-19	-21	-23	-30
-2.4	-30	-24	-21	-20	-18	-16	-14	-13	-12	-11	-11	-11	-12	-14	-17	-20	-18	-18	-20	-23	-29
-2.8	-31	-26	-24	-23	-21	-18	-15	-13	-11	-10	-10	-10	-12	-13	-16	-19	-18	-18	-20	-23	-28
-3.2	-33	-28	-25	-24	-24	-20	-17	-14	-12	-11	-11	-13	-14	-17	-20	-19	-19	-21	-24	-29	
-3.6	-35	-30	-27	-26	-26	-23	-19	-17	-15	-14	-13	-14	-15	-17	-19	-22	-21	-22	-23	-26	-32

Figure 10a. 0° difference in phase

Detailed Printout

Phase 1 = 330.000 degrees, Amplitude 1 = 0.741 (-1.3001 dB)
Phase 2 = 0.000 degrees, Amplitude 2 = 0.661 (-1.8000 dB)
Phase 3 = 0.000 degrees, Amplitude 3 = 0.537 (-2.7003 dB)
Phase 4 = 330.000 degrees, Amplitude 4 = 1.000 (0.0000 dB)

Azimuth is horizontal, and elevation vertical.

Comments: Levels in dB below reference, -3dB and -10 dB contours indicated

	-4.0	-3.6	-3.2	-2.8	-2.4	-2.0	-1.6	-1.2	-0.8	-0.4	0.0	0.4	0.8	1.2	1.6	2.0	2.4	2.8	3.2	3.6	4.0
3.6	-36	-30	-26	-23	-22	-22	-27	-22	-18	-16	-14	-14	-15	-16	-17	-21	-26	-27	-26	-28	-33
3.2	-33	-26	-22	-19	-18	-18	-21	-19	-14	-12	-11	-11	-11	-12	-14	-17	-22	-26	-24	-25	-29
2.8	-31	-24	-20	-17	-15	-15	-16	-18	-14	-11	-10	-9	-10	-10	-12	-15	-18	-27	-24	-25	-28
2.4	-30	-23	-18	-16	-14	-12	-12	-14	-16	-14	-11	-10	-10	-10	-11	-13	-15	-19	-26	-30	-30
2.0	-29	-23	-19	-16	-14	-12	-12	-12	-13	-14	-16	-20	-17	-16	-19	-15	-17	-20	-24	-30	
1.6	-26	-21	-18	-17	-15	-12	-10	-9	-7	-7	-6	-7	-8	-10	-13	-16	-14	-14	-16	-19	-25
1.2	-24	-19	-16	-14	-14	-13	-10	-7	-5	-4	-3	-4	-4	-6	-8	-11	-13	-13	-14	-16	-22
0.8	-23	-18	-14	-12	-12	-12	-9	-6	-4	-2	-2	-2	-3	-4	-6	-9	-12	-12	-12	-15	-20
0.4	-22	-16	-13	-11	-10	-11	-9	-5	-1	0	-1	-1	-1	-2	-4	-7	-12	-11	-12	-14	-19
0.0	-22	-16	-12	-10	-9	-10	-10	-5	-1	0	0	-1	-1	-2	-4	-7	-12	-11	-12	-14	-19
-0.4	-22	-16	-12	-10	-9	-9	-10	-6	-2	-1	-1	-1	-1	-2	-4	-8	-12	-12	-12	-15	-19
-0.8	-24	-17	-13	-11	-9	-10	-11	-8	-5	-2	-2	-2	-2	-4	-5	-9	-13	-14	-14	-16	-20
-1.2	-25	-19	-14	-12	-10	-10	-12	-10	-7	-5	-4	-4	-4	-5	-7	-11	-14	-17	-16	-18	-22
-1.6	-27	-20	-16	-14	-12	-12	-12	-10	-8	-7	-7	-7	-8	-10	-15	-17	-20	-20	-21	-26	
-2.0	-28	-22	-18	-15	-14	-14	-14	-13	-11	-10	-10	-10	-11	-13	-18	-18	-20	-22	-24	-30	
-2.4	-30	-24	-21	-19	-18	-17	-15	-14	-12	-11	-10	-11	-11	-13	-15	-17	-20	-19	-21	-23	-29
-2.8	-31	-25	-22	-21	-20	-19	-18	-16	-14	-12	-11	-10	-11	-12	-14	-17	-20	-19	-20	-23	-28
-3.2	-32	-27	-24	-22	-21	-21	-18	-15	-13	-12	-11	-11	-12	-13	-15	-18	-21	-21	-22	-24	-29
-3.6	-35	-29	-26	-24	-23	-24	-21	-18	-15	-14	-13	-13	-14	-15	-17	-20	-24	-23	-24	-27	-32

Figure 10b. 30° difference in phase

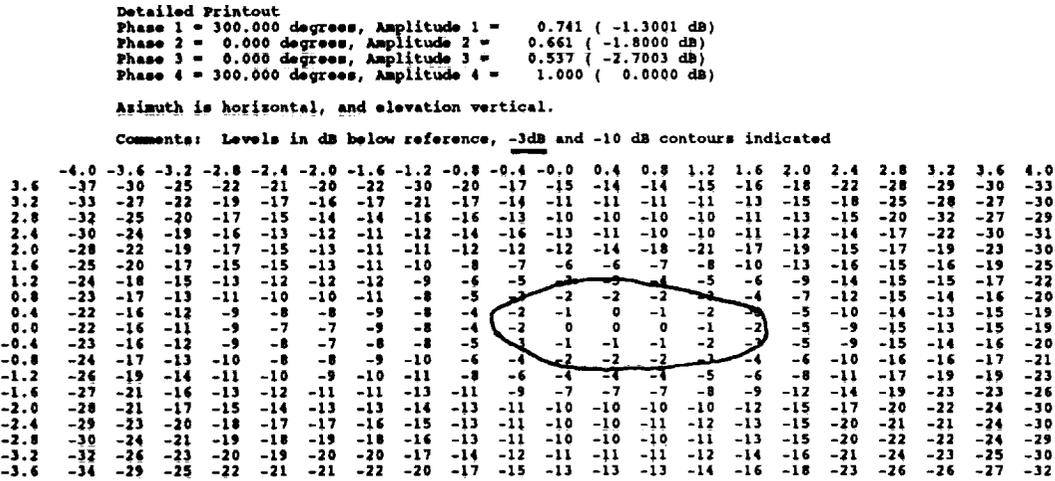


Figure 10c. 60° difference in phase

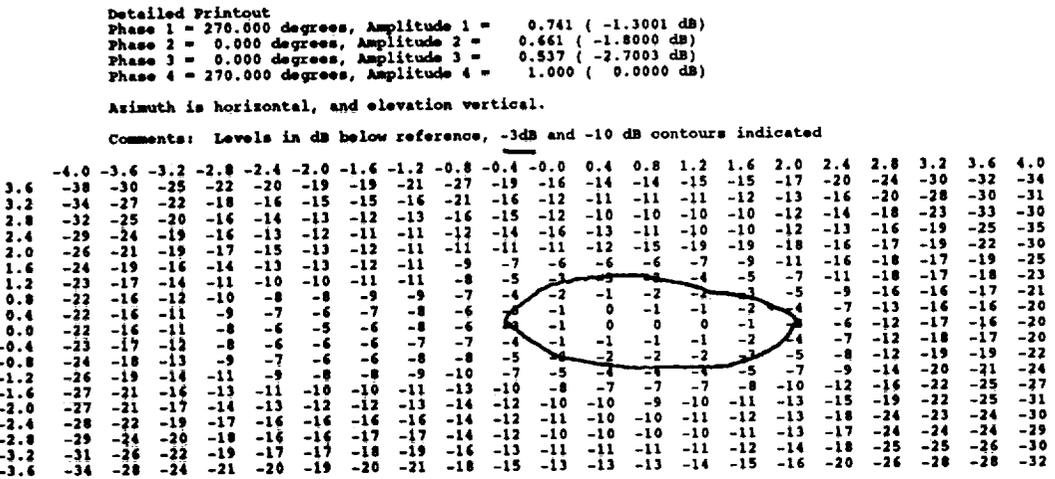


Figure 10d. 90° difference in phase

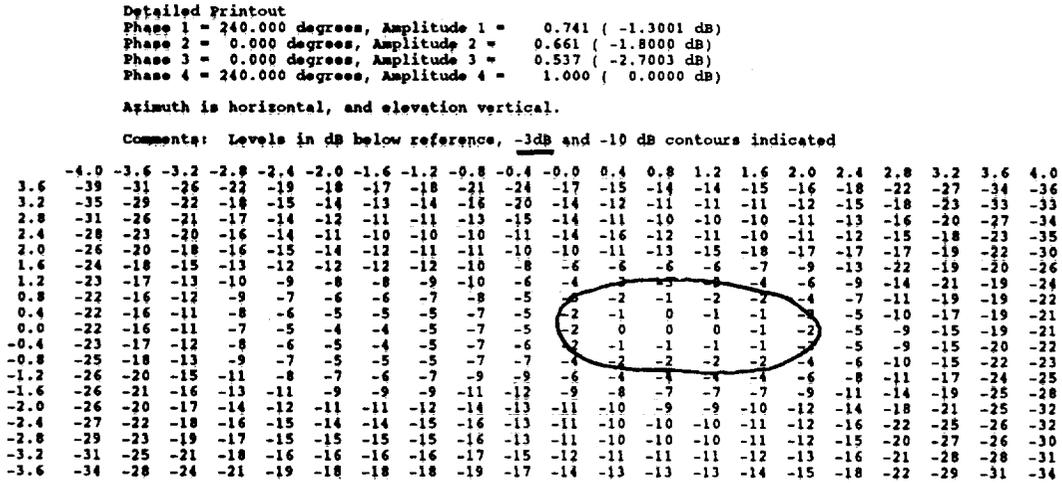


Figure 10e. 120° difference in phase

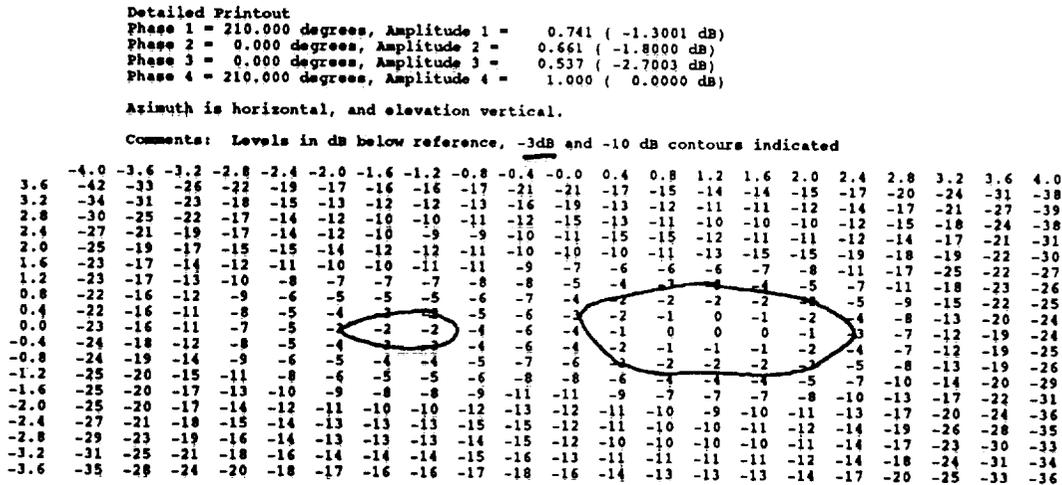


Figure 10f. 150° difference in phase

Detailed Printout
Phase 1 = 180.000 degrees, Amplitude 1 = 0.741 (-1.3001 dB)
Phase 2 = 0.000 degrees, Amplitude 2 = 0.661 (-1.8000 dB)
Phase 3 = 0.000 degrees, Amplitude 3 = 0.537 (-2.7003 dB)
Phase 4 = 180.000 degrees, Amplitude 4 = 1.000 (0.0000 dB)

Azimuth is horizontal, and elevation vertical.

Comments: Levels in dB below reference, -3dB and -10 dB contours indicated

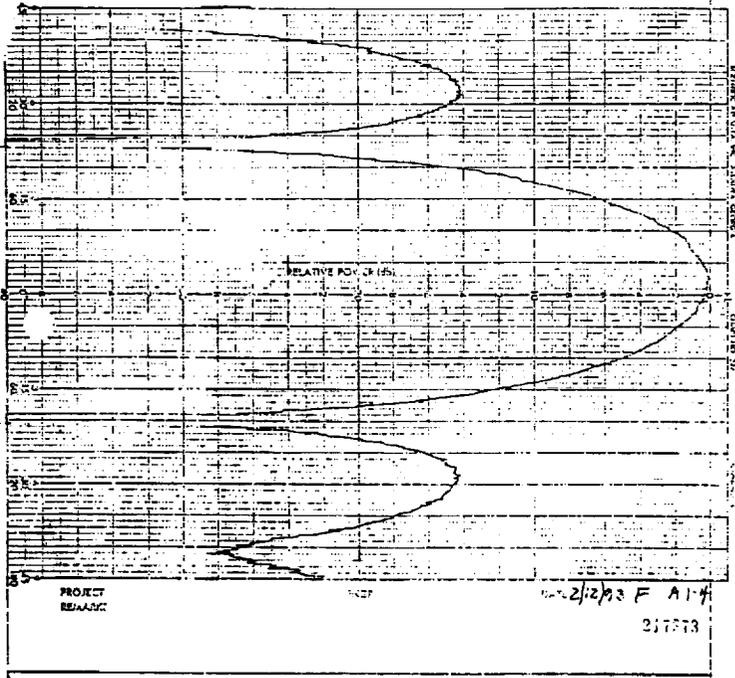
	-4.0	-3.6	-3.2	-2.8	-2.4	-2.0	-1.6	-1.2	-0.8	-0.4	-0.0	0.4	0.8	1.2	1.6	2.0	2.4	2.8	3.2	3.6	4.0
3.6	-39	-39	-28	-22	-19	-17	-15	-15	-15	-17	-21	-20	-16	-15	-14	-15	-16	-19	-23	-28	-36
3.2	-32	-30	-26	-19	-15	-13	-12	-11	-11	-12	-16	-17	-13	-11	-11	-11	-13	-15	-19	-24	-33
2.8	-28	-24	-23	-18	-14	-12	-10	-9	-9	-10	-12	-16	-13	-11	-10	-10	-11	-14	-17	-22	-31
2.4	-26	-20	-18	-17	-15	-12	-10	-9	-8	-8	-9	-11	-16	-14	-12	-11	-12	-14	-16	-20	-28
2.0	-24	-18	-16	-14	-14	-14	-13	-13	-12	-10	-9	-10	-11	-12	-13	-14	-24	-18	-19	-22	-30
1.6	-23	-17	-14	-11	-10	-9	-8	-9	-10	-11	-8	-6	-6	-6	-6	-7	-10	-14	-20	-25	-30
1.2	-23	-16	-12	-9	-7	-6	-5	-5	-6	-8	-7	-5	-4	-4	-4	-6	-9	-14	-22	-32	
0.8	-23	-16	-12	-8	-6	-4	-3	-4	-6	-6	-4	-2	-2	-2	-2	-4	-7	-12	-18	-30	
0.4	-23	-16	-11	-8	-5	-4	-2	-2	-4	-5	-4	-1	-1	0	-1	-2	-6	-10	-16	-28	
0.0	-23	-17	-12	-7	-4	-3	-1	-1	-1	-5	-5	-1	0	0	-1	-2	-5	-9	-15	-27	
-0.4	-23	-19	-13	-8	-5	-4	-2	-1	-2	-5	-4	-2	-1	-1	-1	-2	-6	-10	-15	-26	
-0.8	-23	-19	-15	-10	-6	-4	-3	-2	-3	-4	-6	-5	-3	-2	-2	-4	-7	-11	-16	-27	
-1.2	-23	-19	-16	-11	-8	-6	-5	-4	-4	-6	-8	-7	-5	-4	-4	-4	-6	-9	-12	-17	-29
-1.6	-24	-19	-17	-13	-10	-8	-7	-7	-7	-9	-11	-10	-8	-7	-7	-7	-9	-12	-15	-20	-33
-2.0	-25	-19	-16	-14	-11	-10	-9	-9	-10	-11	-13	-12	-10	-10	-10	-11	-12	-16	-19	-23	-39
-2.4	-27	-21	-17	-15	-13	-12	-11	-11	-12	-14	-14	-12	-11	-10	-10	-11	-13	-17	-22	-28	-38
-2.8	-29	-23	-19	-16	-14	-12	-11	-11	-12	-14	-14	-11	-10	-10	-10	-10	-12	-16	-20	-27	-37
-3.2	-32	-25	-21	-17	-15	-13	-13	-13	-13	-15	-15	-13	-11	-11	-11	-11	-13	-17	-21	-27	-38
-3.6	-35	-29	-24	-20	-17	-16	-15	-14	-15	-17	-17	-15	-14	-13	-13	-14	-16	-19	-23	-29	-39

Figure 10g. 180° difference in phase

IV. Application to the Hardware

The next step was to apply these results to the re-tuning of the pattern for the KTx subsystem, through careful manipulation of the inline mechanical phase shifters. Unfortunately, the power amplifier driving element A2 (A3 in SSG docs) failed, leaving only elements 1, 3, and 4 with which to work with. Applying the expected amount of phase in the 1,4 combination's feed network resulted in the following pattern improvement in Figure 11.

ORIGINAL PAGE IS
OF POOR QUALITY



Pattern 302D, A14

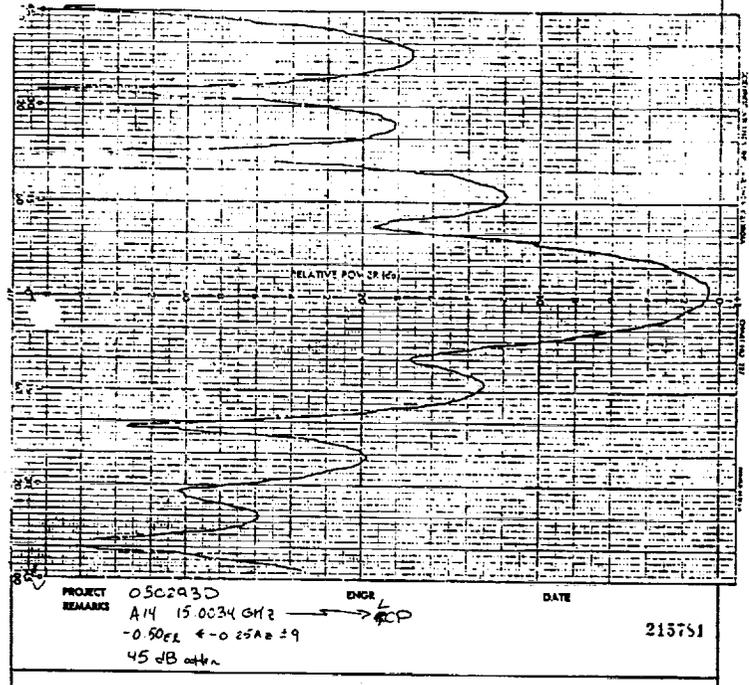


Figure 11. 1 & 4 actual phased pattern⁵

⁵ Taken from TSG-93-75, pg. 4-24, Pattern 212F/302D, March 2, 1993

Returning to ASP for a "14" co-phased estimate reported the following predicted pattern:

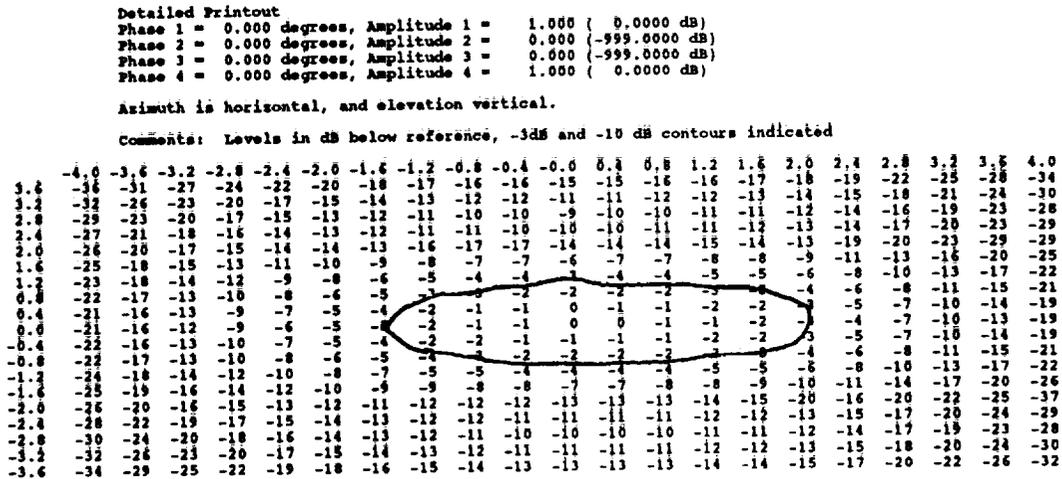


Figure 12. 1 & 4 predicted pattern from ASP

V. Summary

ASP expertly assisted in the diagnosing of phase shifter errors during KTx testing. Additionally, the software was able to correctly estimate the effect of phase shift on the overall array pattern developed from each of the internally held element patterns. It would have been advantageous to work with the complete KTx array, with all elements operating, but due to system failures, this level of analysis was not supported.

VI. Future Applications

A specially modified version of ASP has been used to obtain predictions for XTE solar sail blockage effects, including interferometer peaking due to solar sail reflection. Future applications for Space Network projects may include diffraction effects for communications regions directly blocked by the XTE solar panels, cross polarized pattern predictions, antenna squint determination, axial ratio calculations, pattern normalcy (null suggested beamwidths versus actual beamwidth) and diskette loadable antenna patterns.

Simulation may even be further extended through the addition of TDRSS link simulation code, allowing the user to identify degradation in service due to misphasing or misalignment,

with a standard six-element based set of orbital dynamics calculations.

ASP generates several outputs, including a variable sweep (azimuth and elevation) on-screen display, a KTx specific printed sweep, elevation and azimuth cuts to ASCII files (Lotus and Quattro compatible), and an XTE specific contour plot designed to demonstrate solar panel interference.

Allen David Snook, was born in Fall River, Massachusetts on February 15, 1970. He received the B.S. degree in Electrical Engineering from Virginia Polytechnic Institute and State University. He joined the Loral Aerosys SEAS contract in 1992. His main area of interest is theoretical electromagnetics, and the application of computer and software toward solution of electromagnetics problems.



**Conversion of the Proprietary ROLM Inter-Node Link
from Multimode to Singlemode Operation**

by Larry Boucher
GTE Government Services
White Sands Ground Terminal
(505) 525-6976

SUMMARY

Many NASA centers have selected ROLM®¹ Computerized Branch Exchanges (CBXs) as their standard telephone exchange. The ROLM 9751 CBX Model 70 with ROLM software release 9005 can inter-communicate as a "multi-node" system over a multimode fiber optic link of 450 to 6,000 meters. Singlemode fiber installations are not supported by ROLM. Two New Mexico-based NASA satellite ground terminals were already connected via a 6 kilometer singlemode fiber optic link. The ROLM Inter-Node Link (INL) was converted from multimode LED transmitters to singlemode laser transmitters and two ROLM CBX systems were interconnected using the modified INL. On activation, the system operated normally and has done so for six months. System testing indicates sufficient margin to drive 45 kilometers of singlemode fiber, an important benefit for widely separated facilities.

¹ ROLM® is a registered trademark of ROLM Systems

I. INTRODUCTION

The NASA White Sands Complex (WSC) is a tenant of the Department of Defense (DoD) on the White Sands Missile Range (WSMR). The site consists of a space transportation testing facility operated by the Johnson Space Center (JSC) and two Tracking and Data Relay Satellite System (TDRSS) ground terminals operated by Goddard Space Flight Center (GSFC). For external telephone service, the original tenant, White Sands Test Facility (WSTF) is equipped with a fiber optic Main Point-of-Presence (MPOP) from the local exchange carrier (LEC), US West. Copper and fiber optic facilities serve the site telephony requirements from the MPOP.

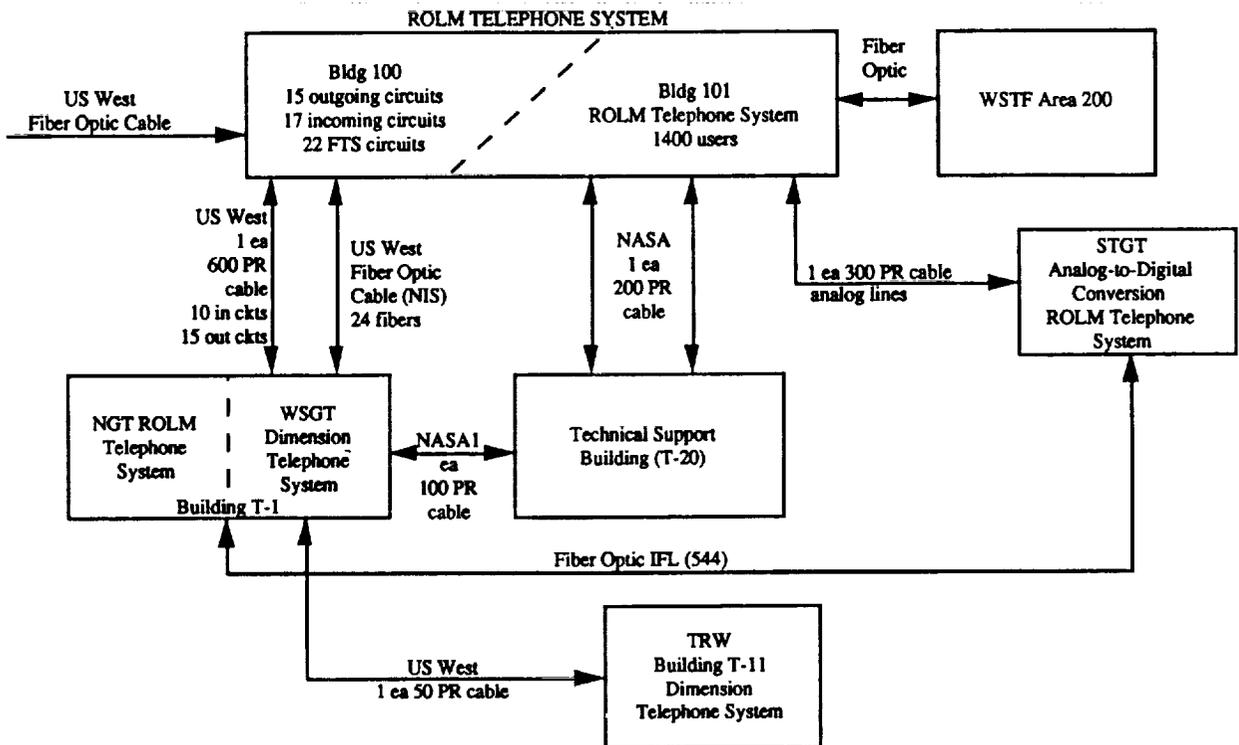
The original TDRS ground terminal, White Sands Ground Terminal (WSGT), was provided a 600 pair copper facility from WSTF and used an analog Dimension^{®2} branch exchange for site telephony. In 1989, NASA constructed the Second TDRS Ground Terminal (STGT) and selected a ROLM 9751 Model 50 with release 9004 software for the branch exchange. By that time, JSC/WSTF had already converted to a ROLM 9751 Model 70 two-node configuration with release 9004 software. STGT was not provided with a point-of-presence by the LEC. NASA contracted US West to install a 6 kilometer long 300 pair copper facility to provide external telephony service via connectivity to the JSC/WSTF ROLM system and the JSC/WSTF MPOP.

As part of an upgrade plan, NASA removed the leased, analog Dimension telephone system and installed a purchased, digital ROLM telephone system at WSGT in March, 1993. The telephone service configuration before these modifications is shown in Figure 1.

A 150MBPS fiber optic MPOP at WSGT was also activated as the external interface. This MPOP was derived from the 280MBPS JSC/WSTF MPOP. The WSC activation plan included installation of a 6 kilometer singlemode fiber optic Inter-Facility Link (IFL) between the two sites. This facility was designed for high data rate transfers of satellite data.

The STGT ROLM system was not capable of receiving high capacity T1 circuits over the 6 kilometer copper facility. Telephone use at STGT was increasing as the station integration and test activity proceeded. Additional capacity through the JSC/WSTF ROLM system was limited. It was decided to interconnect the two TDRSS facilities in a ROLM multi-node configuration.

² Dimension[®] is a registered trademark of AT&T



White Sands Complex Telephone Systems Before Upgrade
Figure 1

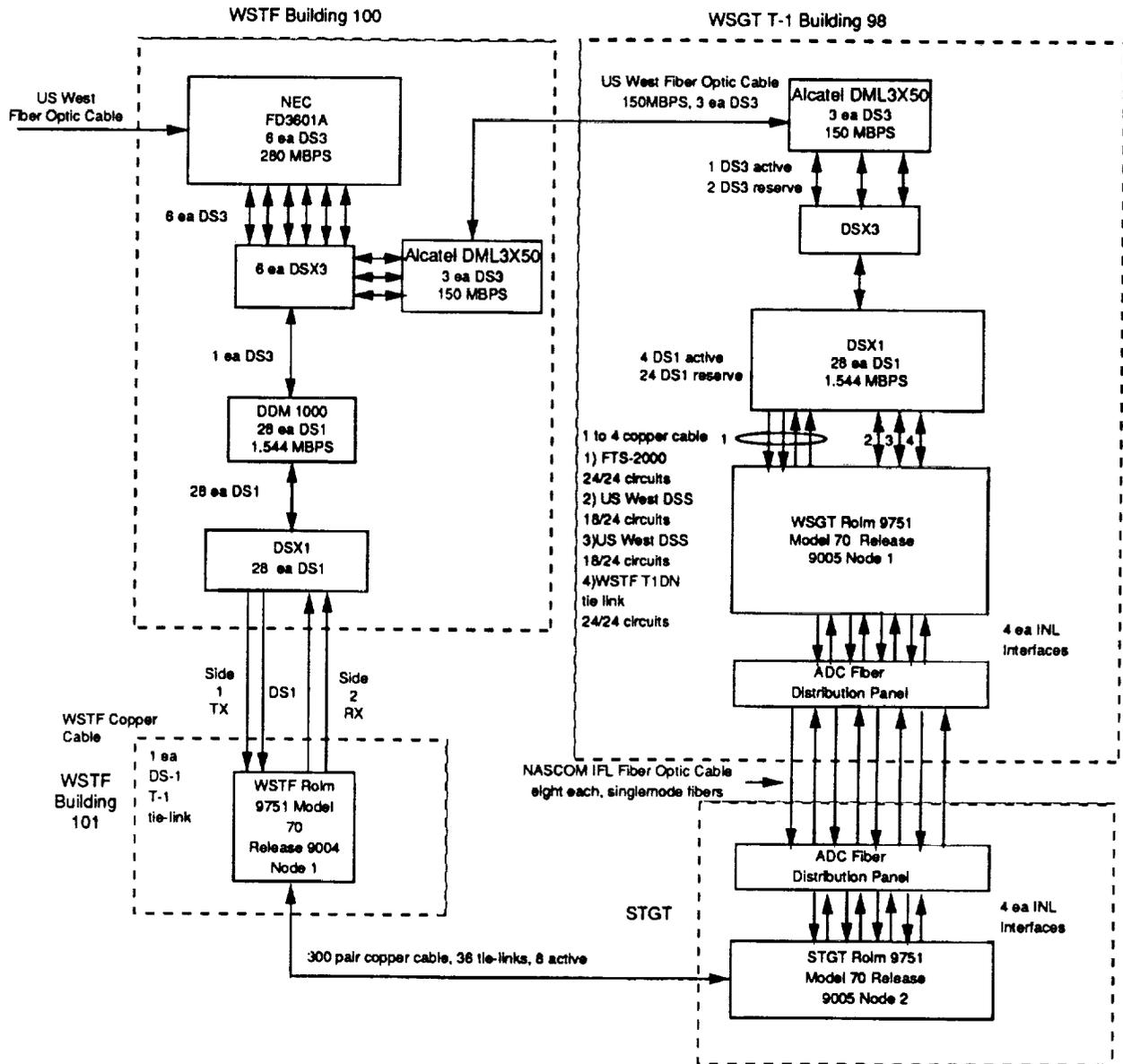
This required conversion of the STGT ROLM system from a 9751 Model 50 to a 9751 Model 70 and from software release 9004 to 9005 along with installation of a suitable distributed node connection between the two CBX systems. As a multi-node configuration STGT would be able to share trunking services from the point-of-presence with WSGT; the two CBXs would be redundant to each other increasing the system availability.

Five implementation options were considered to interconnect the two ROLM CBX systems: 1) installing a multimode fiber optic cable between the two sites to support the ROLM multimode INL, 2) installing subscriber carrier equipment to carry high capacity circuits over an existing copper link, 3) using external multimode to singlemode conversion equipment for ROLM INL connectivity via the singlemode fiber, 4) multiplexing T1 circuits for singlemode fiber transmission, and 5) converting the ROLM INL Fiber Optic Extender (FOX) hardware to drive singlemode fibers.

After considering the costs, advantages, disadvantages, and implementation risks of each of these methods, it was decided to convert the ROLM hardware to a non-standard configuration to drive the available IFL singlemode fibers directly. The advantages were lowest cost and easiest installation. The disadvantages were no factory support for the modified configuration and a higher health (vision) risk from the high energy output lasers. This option assumed an operability and reliability risk since the conversion had never been done before. The risk was mitigated by contracting the ROLM Company for engineering support: bench testing the selected laser transmitter with their proprietary INL protocol and field performance testing the modified system. The installed configuration is shown in Figure 2.

II. SYSTEM DESCRIPTION

The ROLM 9751 Model 70 Computerized Branch Exchange is a digital telephone private exchange capable of supporting up to 1045 simultaneous voice or data connections per node [1]. A multi-node 9751 CBX has from two to fifteen nodes functioning as a single system. Three multi-node categories are supported based on the distance between nodes. "Collocated nodes" are up to 61 meters apart and use twinaxial cable for interconnection. "Distributed nodes" are 61 to 6,000 meters apart. When the node separation is up to 300 meters, twinaxial cable is used. When the separation is up to 450 meters, IBM Type 2 cabling is used. For separations



White Sands Complex Telephone Systems After Modification
Figure 2

of up to 6,000 meters, the proprietary ROLM Inter-Node Link (INL) interface is specified. Multimode fiber is used for INL node-to-node connectivity. "Remote nodes" are separated by distances between 6 kilometers and 80 kilometers. Remote nodes use a different ROLM interface known as the Extended Digital Intertie (XDI) operating over standard T1 transmission facilities [2]. The WSC application was at the upper boundary for distributed nodes at 6 kilometers between tracking stations.

The INL operates at 74MBPS and supports 545 channels per link. This is more than sufficient intersite capacity. The system design is robust including a primary and a secondary fiber optic link. Each of the links is redundant. Eight fibers are required for full connectivity: four transmit and four receive fibers. Specifications for the interconnecting fiber optic facility [3] recommend multimode, graded index fiber with 62.5μ core diameter and 125μ cladding diameter (other multimode fibers of between 50μ and 100μ core sizes can also be used). A maximum end-to-end optical power loss of up to 13dB is acceptable for the recommended 62.5μ core diameter fiber. Specifications call for a source central wavelength of 1310 nm.

The ROLM fiber optic extender (FOX) printed wiring board uses a light-emitting diode (LED) multimode transmitter module and a positive-intrinsic-negative (PIN) diode receiver module. Both of these modules are manufactured by the Sumitomo Electric Fiber Optics Corporation and are designed for a central wavelength of 1310 nm.

III. FIBER OPTIC INTER-FACILITY LINK

The ROLM 9751 multi-node installation decision included the use of an existing fiber optic cable between WSGT and STGT for an Inter-Facility Link (IFL). The IFL has the primary mission of transferring user spacecraft data between sites for multiplexing and retransmission to the user's Payload Operations Control Center (POCC) via the NASA Communications System (NASCOM). This mission requires fiber that can support high data rates with low loss.

The NASCOM IFL specifications [4] are for 6,000 meter bundles of 144 singlemode fibers. The core diameter of each fiber is 8.3μ with 125μ diameter cladding. The fiber is optimized for dual-window central wavelengths of 1310 nm and 1550 nm. The maximum attenuation is 0.50 dB per kilometer at 1310 nm and 0.40 dB/km at 1550 nm. Each fiber is terminated at each end by fusion splicing into a fiber optic pigtail. The pigtails are part of a patch panel with "ST"

type connectors. The specified total link loss (including 6 dB of reserve and future maintenance loss) was ≤ 11.9 dB. The IFL Channel Analysis [5] gives a maximum loss of 5.9 dB for the typical fiber. The IFL has spare fibers. Some of them were allocated for intersite telephony.

The IFL termination racks are located in technical equipment rooms at each site, approximately 50 meters from CBX equipment rooms. A plenum-rated bundle of twelve singlemode fibers was routed from IFL termination patch panels to CBX equipment rooms at the respective sites with patches and splices made to complete an eight fiber link (four spares). At the completion of this task, an optical time domain reflectometer (OTDR) was used to measure the span. A typical OTDR trace is shown in Figure 3. Results from the OTDR measurements for eight operational INL fibers are summarized in Table 1. The extensions added less than one dB to the total loss of the span. This facility had all the qualities to support intra-site telephony except for being the incorrect type of fiber for the ROLM INL. The IFL telephone extension is shown in Figure 4.

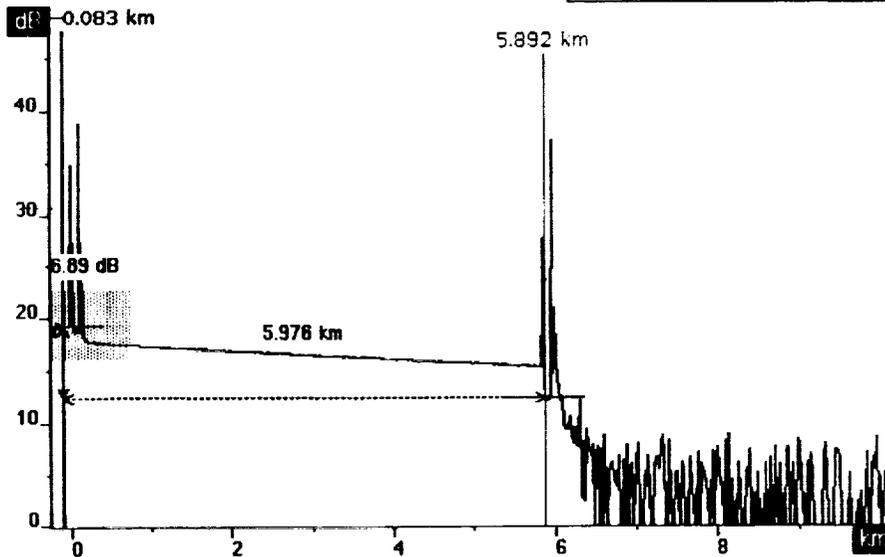
Table 1
Optical Loss for Extended Inter-Facility Link Fibers

Fiber Number	Loss, dB/km	Optical Length, km	Total loss, dB
1	1.055	5.976	6.30
2	0.684	5.076	4.09
3	0.661	5.976	3.95
4	0.605	5.976	3.62
5	0.580	5.976	3.47
6	0.703	5.976	4.20
7	1.152	5.976	6.88
8	0.899	6.034	5.42

Tektronix
TTP2 FiberMaster

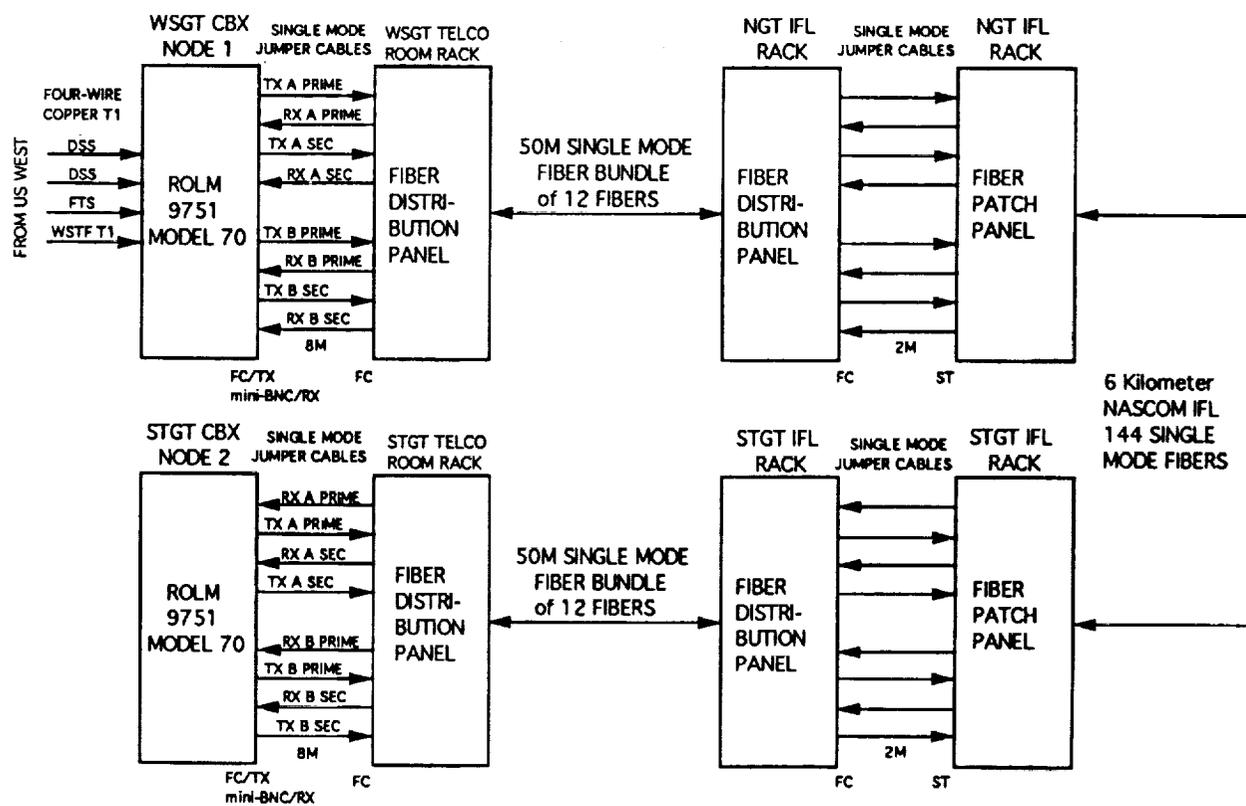
Normal Density

Loss = 1.152 dB/km



Current Waveform	Acquired: 2/09/93 - 18:59:21
Normal Density	Averages = 8192 (0:00:39)
Source = Acquisition	
Module = 1310SM (L1) - b030127	Refractive Index = 1.4710
Pulse Width = 50 ns (5 m) [HR]	Slope Calculation = Two Point
Maximum Range = 10.0 km	
Masking Type = Off	
Fiber Id =	Operator =
Notes =	
Firmware version = T3.13A	

Typical OTDR Trace for the Extended Interfacility Link
Figure 3



Inter-Facility Link Telephone Extensions
Figure 4

IV. ENGINEERING EVALUATION OF CONNECTIVITY OPTIONS

Five options were evaluated before making the decision to modify the factory standard INL configuration to directly support singlemode fiber transmission. Each had advantages, disadvantages, risks, and associated implementation costs.

1. Multimode Fiber Installation

ROLM factory support for INL connectivity requires a multimode fiber communication link, preferably one with a 62.5μ core diameter. This option had the least risk; it was the factory-recommended approach. NASA had already incurred the expense of installing the IFL. This installation was complete and there was no possibility of adding multimode fibers. The costs for trenching in a new fiber were too high to be given serious consideration.

2. Subscriber Carrier Equipment Installation

Subscriber carrier equipment multiplexes individual channels into a data stream carried over copper transmission facilities. This option is like the Remote Node method except it uses copper links. Disadvantages were 1) high cost, 2) lack of ROLM support, 3) use of copper transmission facilities, and 4) equipment self-maintenance. This option also did not receive serious consideration.

3. External Multimode to Singlemode Conversion

Commercially available equipment that could 1) receive multimode fiber optic transmissions for conversion to singlemode and 2) receive singlemode transmissions for conversion to multimode received considerable attention. Numerous fiber optic equipment vendors were contacted. This option had high risk; none of the converter vendors would guarantee success. ROLM would not support this installation and warranty INL functionality. In-house maintenance was required for additional configuration items.

The problem was bit transition density. The format of ROLM INL protocol is proprietary and the specification for bit transition density for transmitter and receiver modules [6] suggested a high duty cycle was required. One vendor intended to support a trial of his equipment, but this option was rejected before a trial could take place. The cost was higher than conversion of the INL to singlemode operation. (After completion of our modification, ROLM Engineering indicated that they were planning further evaluation of external conversion.) This option remains viable.

4. Extended Digital Intertie Interface

ROLM requested that the two nodes be connected using an XDI interface with multiple T1 services. Each T1 has a 24 circuit channel capacity. To emulate the performance of the INL, 23 T1s would have to be multiplexed and carried via the XDI. Four T1s were needed to handle the expected intersite telephony traffic. Since several vendors sell commercially available multiplexer/demultiplexer equipment; this option was very viable. Most of the vendors could support the singlemode IFL and provide the multiplexing capacity that was initially needed with expansion potential by adding modules. This option was almost selected. It had higher cost than option 3, but was ROLM supported. Disadvantages included self-maintenance of transmission equipment and lower channel capacity than the INL. Because the multiplexing and transmission equipment was well proven in numerous field installations by commercial telephone companies and was ROLM supported, this option had low risk.

5. Conversion of ROLM INL to Singlemode Operation

This option had the highest risk. The ROLM INL had never before been operated over singlemode fibers and the factory recommended against it. They would not support or guarantee INL performance. The modified fiber optic extender (FOX) printed wiring boards would not be under warranty nor vendor repaired if they failed. ROLM objected to using laser modules on their boards because of Underwriter Laboratory certification. The LED multimode module was UL certified, but the laser module was not. NASA has configuration management procedures for "altered items," so factory objections were not sufficient for rejection. The UL safety issue could be managed by placarding modified equipment with warning labels.

The fiber optic transmitter and receiver module vendor had a pin-for-pin replacement singlemode laser module. This part was expensive, but the overall cost for direct conversion was much less than any other option. The IFL transmission facilities to use the direct conversion were already in place. Discussions with the vendor indicated that the multimode receiver would operate with output from singlemode transmission facilities. The vendor had a concern of overdriving, or saturating, the receiver module. (Which could have been solved with in-line optical attenuators.) The remaining risk issue was performance. This was mitigated by contracting with ROLM for engineering support. Bench testing the laser transmitter module with the stock receiver module and ROLM INL protocol was performed before altering the FOX

printed wiring boards. Bit-error-rate testing indicated the transmitter module's electrical input would be compatible with the printed wiring board's ECL output logic and that the receiver module worked with optical input from a singlemode laser. The decision was made to convert the ROLM INL from multimode to singlemode fiber optic operation.

V. MODIFICATION CONSIDERATIONS

The FOX printed wiring boards were modified after bench testing. Each one had the LED multimode transmitter module removed and replaced by a laser singlemode module. Each altered board was remarked with a new part number for configuration management. The boards were marked with a warning label for "Laser Output." The modified FOX boards were reinstalled into ROLM CBX INL shelves and were connected to the extended IFL via suitable singlemode fiber optic patch cables. The extended fiber installation had "Laser Output" warning labels located where a connection presented a potential vision hazard. The modification was straightforward and required no special precautions besides electrostatic discharge prevention.

1. Multimode Transmitter Characteristics

The standard INL multimode fiber optic transmitter module is a Sumitomo DMT-54 Optopia^{®3} module (also listed as DM-54-TA). The DMT-54 transmits digital data signals at rates from 1MBPS to 125MBPS in non-return to zero (NRZ) formats. The transmitter uses an InGaAsP LED light source and launches -17 dBm into 62.5 μ /125 μ fibers. The electrical input is differential ECL logic and is certified for data duty cycles of 45 to 55%. The optical output connector is a mini-BNC [7].

2. Multimode Receiver Characteristics

The standard INL multimode fiber optic receiver module is from the same Sumitomo Optopia family. It is the DMR-54 module (also listed as DM-54-RA). It operates at the same data rates and duty cycle as the transmitter. It uses an InGaAs PIN photo detector receiver and has a differential ECL output. The optical input connector is a mini-BNC. The receiver has an optical input sensitivity, or carrier detection level, of -34 dBm with input from a 62.5 μ fiber [8]. The input power saturation level is -8 dBm [9]. Dynamic range is 26 dBm.

³ Optopia is a registered trademark of Sumitomo Electric Industries, Ltd.

3. Singlemode Transmitter Characteristics

The Sumitomo DM-91-TD is a high speed fiber optic laser transmitter module. It supports digital data transmission over singlemode fibers with NRZ data rates ranging from 1MPBS to 250MBPS. It uses an InGaAsP FP laser diode output with a peak emission wavelength of 1.3 μm and a launch power of ≥ -10 dBm. The electrical input is differential ECL. It is a pin-for-pin replacement for the DMT-54. The optical output connector is an "FC" [10].

4. Transmitter Replacement Considerations

Several testing and verification objectives for the modification were established. A test plan to validate these objectives [11] was prepared and testing was conducted to assure that the DM-91-TD laser transmitter module could replace the DMT-54:

- Verify that the laser transmitter module is physically and electrically compatible with the LED transmitter module.
- Verify that a simulated link is functional and reliable; i.e., no bit transition pattern dependency and a bit error rate under 1×10^{-12} at minimum and maximum line attenuation for the INL data rate of 74MBPS.
- Measure system attenuation and attenuation margin.
- Verify pseudo-random data performance in INL operation.

VI. BENCH TEST AND VERIFICATION RESULTS

Bench tests by ROLM verified the physical and electrical compatibility of the DM-91-TD module with the FOX printed wiring board. The replacement module needed to be a form, fit, and functional replacement. This included the physical size and the power requirement from the on-board 5.2 VDC ECL power supply.

1. Physical and Electrical Compatibility

A) Physical Compatibility

The DM-91-TD has the same 24-pin package as the DMT-54. The output connector is an "FC" instead of the mini-BNC. This is a preferable connector, especially for singlemode fibers, since it is noted for good optical alignment and low loss. Singlemode patch cables with FC connectors are low cost off-the-shelf items.

B) Power Consumption

The DM-91-TD current requirement from the -5.2 VDC ECL power supply was measured to be 120 ma, which was 30 ma less than the current requirement for the DMT-54,

2. Bit Error Rate Testing

Bit error rate testing with different data patterns will show if there are pattern dependencies and attenuation ranges for low error rate performance. The BER was $\leq 1 \times 10^{-12}$.

A) Data Pattern Dependency

A BER test series showed that there were no bit pattern or transition density dependencies with pseudo-random data and word lengths of up to $2^{23}-1$ bits.

B) INL Data Rate BER Performance

Tests of the bit error rate performance were made at two data rates: 74MBPS and 125MBPS. For both tests, the BER was less than 1×10^{-12} .

C) Receiver Input Power Performance

The maximum input power for the receiver was -8 dBm. BER performance degraded at higher input power levels. The receiver saturated at -4 dBm.

D) Fiber Optic Cable Attenuation Margin

The simulated fiber optic cable attenuation was varied until the BER increased beyond 1×10^{-12} . The line attenuation must be between -5 dBm and -25 dBm for acceptable BER.

VII. FIELD TEST AND VERIFICATION RESULTS

On installation and activation, the two-node ROLM 9751 CBX system operated normally with the modified INL. ROLM diagnostic tests revealed no problems. Field testing (with assistance from the ROLM Company) verified performance margins and provided additional customer assurance.

1. Optical Power Margin Test Results

Optical power measurements assured that each component of the system was good and quantified the optical losses through the entire link. Table 2 summarizes these results. The apparent discrepancies have been attributed to the optical alignment and coupling of the power meter cable. Optical output of the transmitters ranged from -3.6 to -6.9 dB. The average measured optical loss through the system was -6.3.

Table 2
Optical Power Measurements for Modified INL

Fiber Number	Node 1 CBX dB	Node 1 IFL dB	Node 2 IFL dB	Node 2 CBX dB	Total Loss dB	OTDR dB
1	-3.6 Transmit	-5.1	-9.9	-10.1 Receive	-6.5	-6.30
2	-13.7 Receive	-10.0	-6.7	-5.6 Transmit	-8.1	-4.09
3	-6.9 Transmit	-7.2	-10.5	-11.8 Receive	-4.9	-3.95
4	-9.6 Receive	-7.3	-4.4	-3.9 Transmit	-5.7	-3.62
5	-5.9 Transmit	-6.9	-10.5	-11.4 Receive	-5.5	-3.47
6	-11.5 Receive	-10.2	-6.9	-5.9 Transmit	-5.6	-4.20
7	-4.8 Transmit	-6.0	-11.0	-11.5 Receive	-6.7	-6.88
8	-12.0 Receive	-9.0	-5.6	-4.7 Transmit	-7.3	-5.42
Averages	-5.2 Transmit			-11.5 Receive	-6.3	-4.57

2. Attenuation Margin Test

An optical attenuator was inserted at the receiver end of the active INL link. Attenuation was added until the INL receiver lost lock as indicated by an LED on the FOX board driven from the carrier detect circuitry on the receiver module [12]. Insertion of 20.6 dB of optical loss was required to break the INL lock. The INL specification for fiber attenuation is 0.5 dB/km. The attenuation test would then indicate the this system is capable of driving an

additional 41 km for a total of 47 km. With a one dB margin, the system could drive a 45 km long fiber, or longer fibers if a lower loss cable was used.

3. Pseudo-Random Data Testing

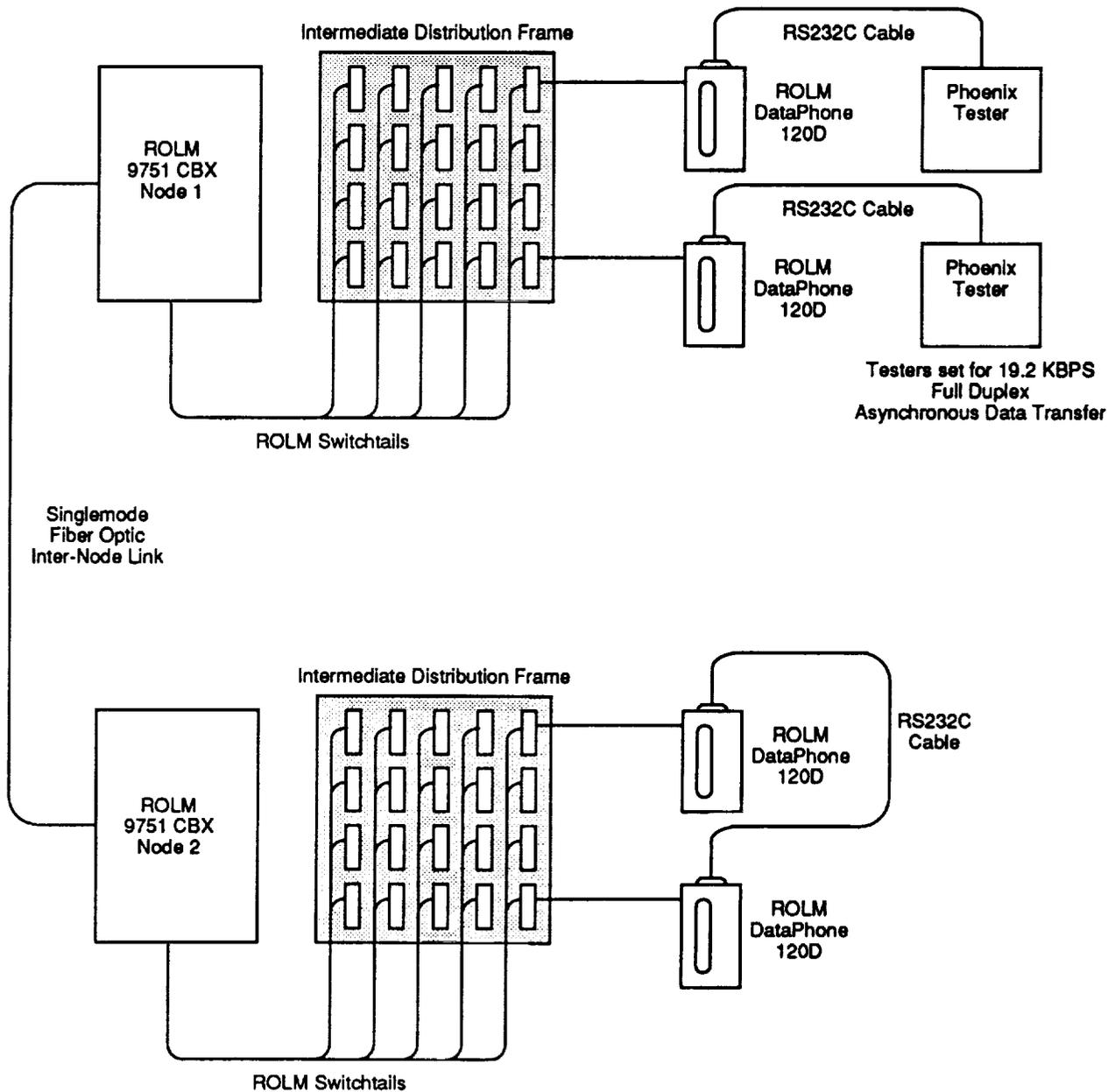
For confidence, the modified INL was tested with pseudo-random data. Two 120D ROLMphones⁴ were configured for data transmissions at each site and full duplex communications were established. The test configuration is shown in Figure 5. Pseudo-random data at 19.2KBPS was sent through the system from one 120D through the pair at the remote end and back to the other 120D. The test was bi-directional. Each 120D was transmitting and receiving simultaneously. This test ran error-free for two days. This was insufficient time at that data rate to establish a bit error rate, but it did add customer confidence for the operational performance of the modified Inter-Node Link. The system has been in service since March 1993 without any operational problem related to the modification.

VIII. CONCLUSIONS

The ROLM 9751 CBX Model 70 with software release 9005 can operate in a multi-node configuration using the ROLM Inter-Node Link over singlemode fibers. Using singlemode optical fibers requires modifying the ROLM Fiber Optic Extender printed wiring boards to accept laser transmitter modules in place of the LED transmitter modules. When installed with a low-loss singlemode span (0.5 dB/km or less), node separation can be up to 45 kilometers. This node separation distance is comparable to that for "remote nodes," yet the channel capacity remains high (545 channels per INL) and the cost is less than for Extended Digital Intertie interfaces. This application is of interest to ROLM users with widely separated facilities or ones with existing singlemode fiber facilities.

Using external multimode to singlemode optical converters for the same function could be investigated further. This was evaluated, but was not pursued for this application. A key advantage is not having to modify and void warranties on the ROLM FOX printed wiring boards. Commercial converters would have UL certification.

⁴ ROLMphone is a registered trademark of ROLM Systems



Test Configuration for Data Transfer over the INL
Figure 5

ACKNOWLEDGEMENTS

The author acknowledges and appreciates support from GTE Government Services in sponsoring this paper. Mr. Ron Sowell was especially encouraging. Mr. Jamal Alrawi from ROLM Engineering performed the bench testing and supported field testing. Mr. Henry Takahashi and Mr. Tak Fukuoka from Sumitomo Electric Industries provided sample laser transmitters, specifications, and expedited overseas delivery of the operational laser transmitters. Ms. Mary Ellen Shoe, NASA Telephone Systems Manager, Code 543, negotiated with ROLM for test support. Mr. Richard Von Wolff, NASA Project Technical Representative, Code 530.1, supported the INL modification decision.

REFERENCES

- [1] ROLM 9750 Business Communications System, System Description, Release 9005, G522-2110-00, June 1990, Table 2-4, Page 2-22
- [2] ROLM 9750 Business Communications System, System Description, Release 9005, G522-2110-00, June 1990, Section 2.8.1 through 2.8.5, Pages 2-24 through 2-34
- [3] Donald M. Tsusaki, "Fiber Optic Cabling for Internode Links Topic Text," IBM Internal Memorandum Number PN230B, 15 July 1986
- [4] A. Huse and Shoemake, C., "Specification for the Interfacility Link," 540-057, Goddard Space Flight Center, January 1991
- [5] NASA White Sands Interfacility Link, "White Sands Interfacility Link Channel Analysis," Critical Design Review, Lightwave Spectrum, Inc., 24 March 1992
- [6] "Specifications for 1300 nm Wavelength 125 MBPS Fiber Optic Transmitter/Receiver Modules (DMT-54/DMR-54)," Specification No. TS-S85D006B, Sumitomo Electric Industries, Ltd., 8 January 1988, pages 5 and 10
- [7] "Specifications for 1300 nm Wavelength 125 MBPS Fiber Optic Transmitter/Receiver Modules (DMT-54/DMR-54)," Specification No. TS-S85D006B, Sumitomo Electric Industries, Ltd., 8 January 1988, Table for TRANSMITTER (DM-54T), pages 4 and 5
- [8] "Specifications for 1300 nm Wavelength 125 MBPS Fiber Optic Transmitter/Receiver Modules (DMT-54/DMR-54)," Specification No. TS-S85D006B, Sumitomo Electric Industries, Ltd., 8 January 1988, Table for RECEIVER (DM-54R), pages 4 and 5
- [9] Table showing "Bit Error Rate with Retimer," for DM-54R, Sumitomo Electric Industries, Ltd., Figure 2/5, FAX dated 10 August 1992
- [10] "Specifications for 1300 nm Wavelength 250MBPS Fiber Optic Transmitter Module (DM-91-TD)," Specification OTM-89022, Sumitomo Electric Industries, Ltd., 18 August 1989
- [11] Jamal Alrawi, "Single Mode Fiber Optics Test Plan," ROLM CBX/General Engineering Hardware Development, Private Communication, January 1993
- [12] "Specifications for 1300 nm Wavelength 125 MBPS Fiber Optic Transmitter/Receiver Modules (DMT-54/DMR-54)," Specification No. TS-S85D006B, Sumitomo Electric Industries, Ltd., 8 January 1988, page 12



ADVANCED CODING AND MODULATION SCHEMES FOR TDRSS

By Linda Harrell, Ted Kaplan, Ted Berman, Susan Chang

Stanford Telecom
 7501 Forbes Blvd
 Seabrook, MD 20706
 (301) 464-8900

ABSTRACT — This paper describes the performance of the Ungerboeck and pragmatic 8-Phase Shift Key (PSK) Trellis Code Modulation (TCM) coding techniques with and without a (255, 223) Reed-Solomon outer code if they are used for Tracking Data and Relay Satellite System (TDRSS) S-Band and Ku-Band return services. The performance of these codes at high data rates is compared to uncoded Quadrature PSK (QPSK) and rate 1/2 convolutionally coded QPSK in the presence of Radio Frequency Interference (RFI), self-interference, and hardware distortions. This paper shows that the outer Reed-Solomon code is necessary to achieve a 10^{-5} Bit Error Rate (BER) with an acceptable level of degradation in the presence of RFI. This paper also shows that the TCM codes with or without the Reed-Solomon outer code do not perform well in the presence of self-interference. In fact, the uncoded QPSK signal performs better than the TCM coded signal in the self-interference situation considered in this analysis. Finally this paper shows that the E_b/N_0 degradation due to TDRSS hardware distortions is approximately 1.3 dB with a TCM coded signal or a rate 1/2 convolutionally coded QPSK signal and is 3.2 dB with an uncoded QPSK signal.

I. INTRODUCTION

TDRSS users are expected to require higher data rates in the future than are currently supported today. It has been suggested that 8-PSK Ungerboeck and pragmatic TCM codes can support data rates that are twice as high as the data rates currently supported by TDRSS in the same bandwidth. This paper presents the results of an analysis which considers the performance of these codes for S-Band and Ku-Band return services in RFI, self-interference, and hardware distortions. The analysis also considers using the (255, 223) Reed-Solomon code that is recommended by the Consultive Committee for Space Data Systems (CCSDS) as an outer code. This code was selected because it can be easily

implemented with TCM and is effective in protecting against burst errors due to RFI. It is also bandwidth efficient since it only requires 14% overhead. Figure 1 shows the channel model used in the analysis.

The analysis only considers high data rate signals since only high data rate signals require the bandwidth efficiency of TCM codes. The lower data rate signals can achieve better performance with less complexity using a rate 1/2 code.

II. BACKGROUND

The design of the coding and modulation functions were performed separately in traditional communication systems. Ungerboeck presented the concept in [1] that the communications performance could be improved without increasing the bandwidth requirements by designing the coding and modulation functions together. He found that the performance of an uncoded QPSK signal in Additive White Gaussian Noise (AWGN) can be improved by coding the signal and increasing the number of phases modulated onto the carrier. Ungerboeck selected the rate 2/3 convolutional code for use with 8-PSK modulation. Essentially this code maps every two data bits into three symbols and then the three symbols select one of eight phases to be modulated onto the carrier. ([1] describes the approach that is to be used to map each of the two data bits into one of the eight phases so that the resulting code is optimum.) This code with Viterbi decoding can achieve a 10^{-5} BER in AWGN with approximately 3 dB less power than is required for an uncoded QPSK signal without any additional bandwidth. Viterbi showed in [2] that another TCM code can be implemented by coding one data bit into two symbols with a rate 1/2 convolutional code and leaving one data bit unchanged, rather than coding both bits with a rate 2/3 con-

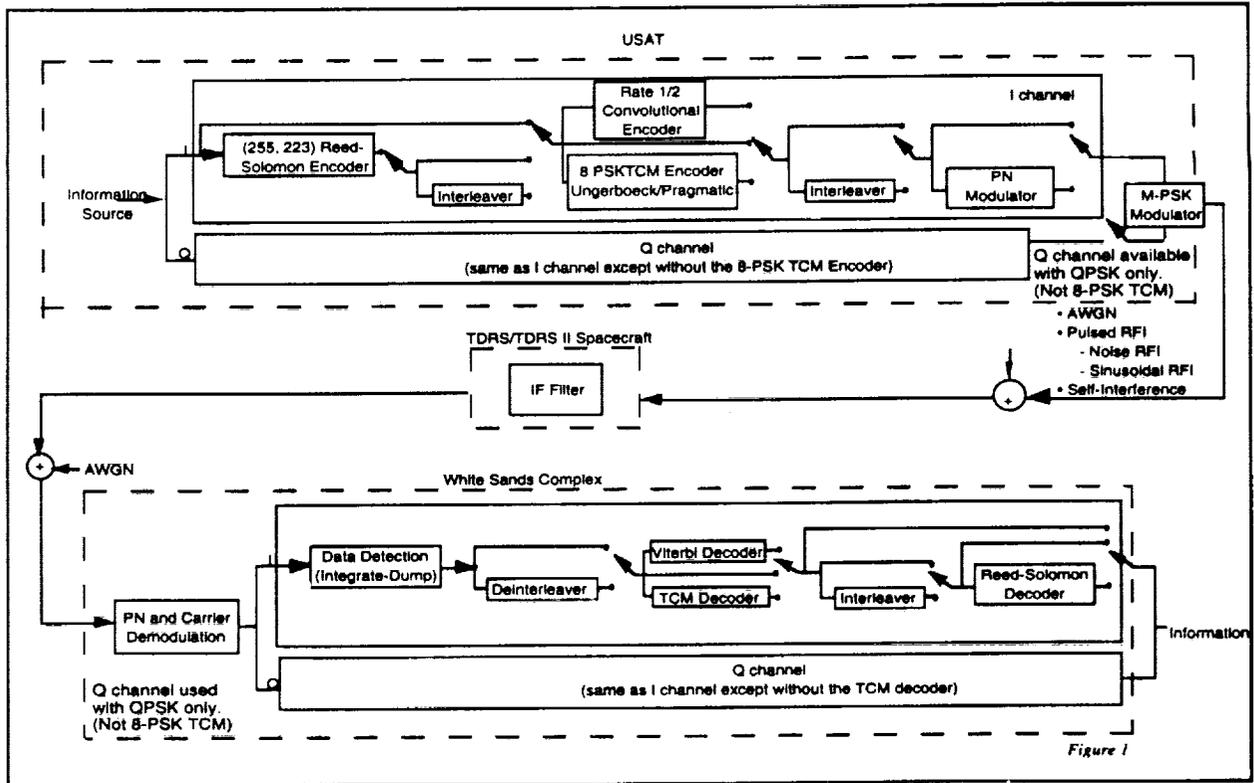


Figure 1. TDRSS Coding and Modulation Channel Model

volutional code. This TCM code, which is referred to as a 8-PSK pragmatic code, is easier to implement and more versatile than the Ungerboeck code, but its performance in AWGN is approximately the same as can be obtained with the Ungerboeck code. Figure 2 shows the BER performance of these two TCM codes in AWGN compared with the performance of uncoded QPSK and rate 1/2 convolutionally coded QPSK.

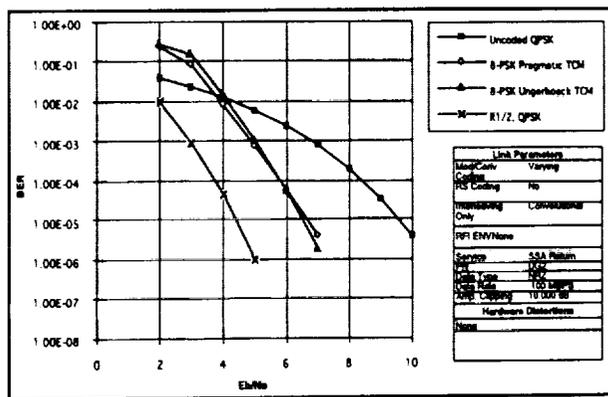


Figure 2. Performance of TCM and Rate 1/2 Convolutional Codes in AWGN

III. ANALYSIS TOOL

The performance of the TCM and Reed-Solomon codes in the presence of RFI, self-interference and hardware distortions was assessed using a Monte-Carlo simulation package that is described in [3].

IV. RESULTS WITHOUT THE OUTER REED-SOLOMON CODE

In the presence of RFI. The performance of the TDRSS return service with the TCM codes was assessed in the presence of the S-Band Multiple Access (SMA) and Ku-Band Single Access (KuSA) terrestrial RFI environments shown in Table 1.

Figure 3 shows the performance of the TCM codes for a 3 Mega bits per second (Mbps) signal in the presence of the SMA RFI environment. This figure shows that the Ungerboeck and pragmatic TCM codes do not perform well in RFI. In fact, the performance of the pragmatic code is not much better than can be achieved with an uncoded QPSK signal. This is because the performance of the pragmatic TCM code is driven by the perfor-

Table 1. RFI Environments for TDRSS SMA and KuSA Return Services

SMA RFI ENVIRONMENT		
5 Pulsed noise RFI sources		
The power spectral density of the noise RFI is uniform within the 6 MHz bandwidth		
Each RFI pulse has a Poisson distributed arrival time and a pulsewidth of 3.5 μ s		
The received power level and duty cycle for each RFI source is as follows:		
RFI Source	P _{rec} Above TDRS Noise Floor (dB)	Duty Cycle (%)
Source 1	35	0.1
Source 2	25	0.4
Source 3	15	1.5
Source 4	5	2.0
Source 5	0	5.0
KuSA RFI ENVIRONMENT		
1 Pulsed sinusoidal RFI source		
The frequency of the sinusoidal RFI is a constant during each pulse, but it changes from pulse to pulse with a probability that is uniform over the 225 MHz channel bandwidth		
Each RFI pulse has a Poisson distributed arrival time		
The received power level of each pulse is 50 dB above the TDRS noise floor		
The RFI duty cycle is 0.1%		

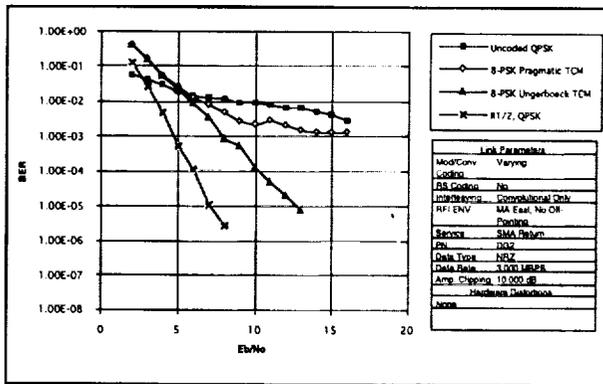


Figure 3. Performance of TCM and Rate 1/2 Convolutional Codes in SMA Return Service RFI

mance of the uncoded bit. The performance of the Ungerboeck TCM code is significantly better than the performance of the pragmatic TCM code because this code does not have any uncoded bits. However, it still suffers about 6 dB degradation at a 10^{-5} BER due to the RFI. (The E_b/N_0 required to achieve a 10^{-5} BER with the Ungerboeck TCM code is approximately 6.6 dB in AWGN and 12.6 dB in the SMA RFI). The problem with the Ungerboeck code is that the rate

2/3 code does not provide sufficient error correction. A lower rate code is needed. A comparison of Figures 2 and 3 shows that the performance with the rate 1/2 convolutional code is only degraded by 2.6 dB due to the RFI. (The E_b/N_0 required to achieve a 10^{-5} BER with rate 1/2 convolutional coding is 4.4 dB in AWGN and approximately 7 dB in the SMA RFI). This explains why rate 1/2 convolutional coding is currently required for TDRSS SMA return links. The SSA return service performance with TCM codes would be degraded by RFI even more than the SMA return service since the SSA RFI environment is even more severe than the SMA RFI environment.

Figure 4 shows the performance of the TCM codes for a 10 Mbps signal in the presence of the KuSA RFI environment. This figure shows that the Ungerboeck and pragmatic TCM codes do not perform much better than the uncoded QPSK in the KuSA return service RFI environment. The coding is unable to correct the errors due to the RFI. However, the rate 1/2 code can correct the errors due to RFI.

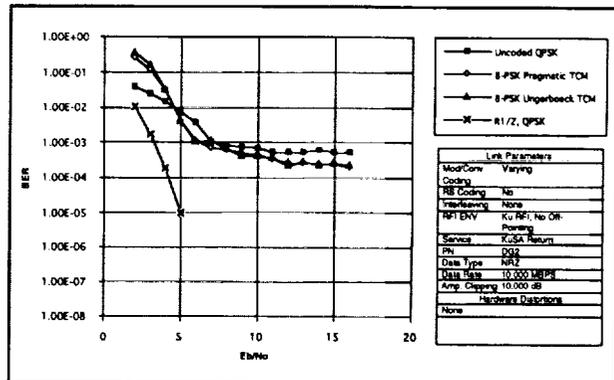


Figure 4. Performance of TCM and Rate 1/2 Convolutional Codes in KuSA Return Service RFI

The performance of the TCM codes in RFI is very important for S-Band return services as RFI can be present for a significant portion of the time a user is in orbit. For example, the SMA return service RFI environment considered in this analysis can be present up to 6% of the time. (This is total time and does not take into account visibility periods.) Ku-Band RFI statistics cannot be generated, but RFI events are much less frequent at

Ku-band than at S-band. This is why TDRSS supports uncoded signals on Ku-Band return links, despite the fact that uncoded signals do not perform well in RFI.

In the presence of Self-Interference. Figure 5 shows the TDRSS return service performance with the TCM codes in the presence of an interfering signal, where the interfering signal has a lower symbol rate than the desired signal. (This is the worst-case situation since none of the interference is filtered at the receiver. But it is also the most likely situation to occur since only the high data rate signals require the bandwidth efficiency of TCM codes.) It was assumed that the desired signal has a 7 dB E_b/N_0 margin, which is sufficient to ensure that noise is insignificant relative to the interference at high signal-to-interference ratios. This figure shows that both the Ungerboeck and pragmatic TCM code's BER performance is worse than the uncoded QPSK signal performance. This is because the decision regions for 8-PSK TCM are closer together than they are for the uncoded QPSK signal.

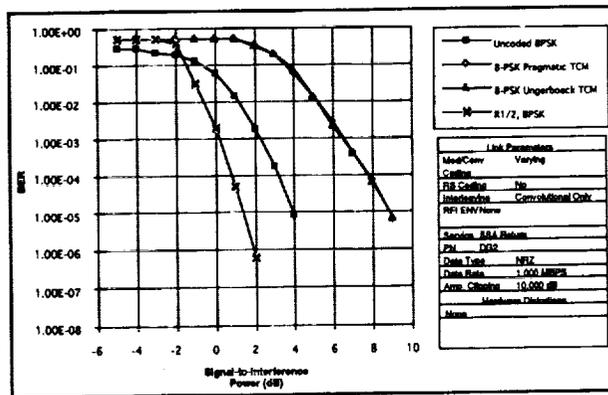


Figure 5. Performance of TCM and Rate 1/2 Convolutional Codes in the Presence of an Interfering Signal with a Lower Symbol Rate than the Desired Signal

The performance of the return services in self-interference is very important for S-Band return services as there are many users operating at the same frequency at S-Band. Self-interference is less of a concern for Ku-Band return services as there are not as many users currently operating at this frequency and beamwidths are narrower at Ku-Band than they are at S-Band. However, self-

interference events are expected to increase as more users operate at the Ku-Band frequency.

In the presence of Hardware Distortions. Table 2 shows the hardware distortions considered in this analysis and the return service performance achievable with the pragmatic TCM coded, uncoded QPSK, and rate 1/2 QPSK signals in the presence of each hardware distortion individually and combined together. This table shows that the E_b/N_0 degradation due to all of the hardware distortions combined is 1.3 dB for the coded signals and 3.2 dB for the uncoded QPSK signal.

Table 2. E_b/N_0 Degradation due to Hardware Distortions

Hardware Distortion		Degradation at a 10^{-5} BER (dB)		
Name	User Constraint Value	Uncoded QPSK	8-PSK TCM Pragmatic / Ungerboeck	Rate 1/2 Coded QPSK
Data Asymmetry	3%	0.8	0.3	0.2
Data Jitter	0.1%	0	0	0
Gain Imbalance	.25 dB	0.2	0.2	0.2
Phase Imbalance	6°	0.6	0.5	0.1
Phase Noise	3°	0	0.2	0
Phase Nonlinearity	3°	0.6	0.1	0.1
AM/AM	0.75 dB/dB	0.3	0	0
AM/PM	12°/dB	0	0	0
Jitter Rate	0.1	0	0	0
Gain Flatness	0.3 dB	0.7	0.1	0.1
All Hardware Distortions Combined		3.2	1.3	1.3

V. RESULTS WITH THE OUTER REED-SOLOMON CODE

In AWGN. Figure 6 shows the performance of the concatenated codes using a (255, 223) Reed-Solomon outer code and the TCM codes and rate 1/2 convolutional code as the inner code in the presence of AWGN. This figure also shows the performance of the (255, 223) Reed-Solomon outer code by itself in AWGN.

In the presence of RFI. Figure 7 shows the performance of the concatenated codes with a (255, 223) Reed-Solomon outer code and either the TCM code or the rate 1/2 convolutional code as the inner code in the presence of SMA RFI environment. A comparison of this figure with Figure 6 shows that the concatenated code performance with the Ungerboeck or pragmatic TCM

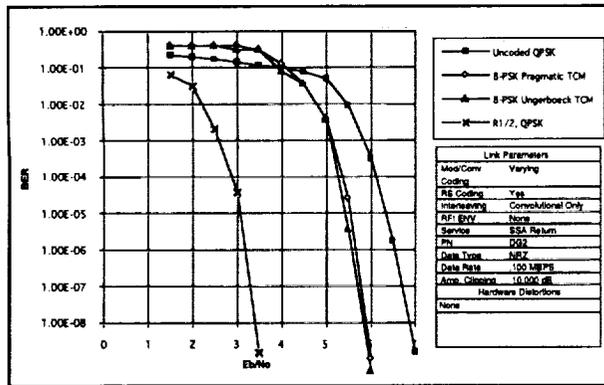


Figure 6. Performance of (255, 223) Reed-Solomon Outer Code with Various Inner Codes in AWGN

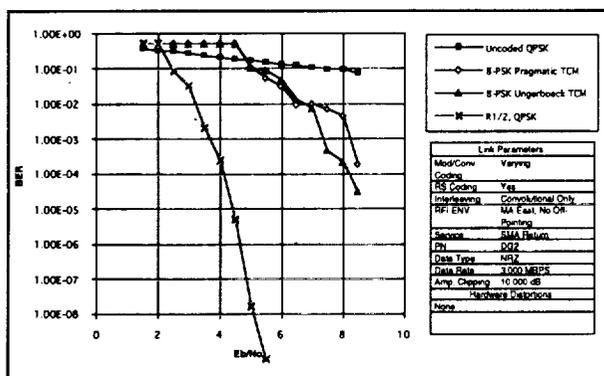


Figure 7. Performance of the Reed-Solomon Outer Code with Various Inner Codes in SMA Return Service RFI

inner code suffers about 3.2 dB degradation at a 10^{-5} BER due to the RFI. (The E_b/N_0 required to achieve a 10^{-5} BER with the Ungerboeck TCM code is approximately 5.5 dB in AWGN and 8.7 dB in the SMA RFI). A similar comparison shows that the performance with the rate 1/2 convolutional code is only degraded by 1.3 dB due to the RFI. (The E_b/N_0 required to achieve a 10^{-5} BER with rate 1/2 convolutional coding is 3.1 dB in AWGN and approximately 4.4 dB in the SMA RFI). Therefore, the concatenated code using either a TCM inner code or a rate 1/2 convolutional code can achieve the required 10^{-5} BER with an acceptable amount of degradation in the presence of RFI.

Figure 8 shows the performance of the concatenated codes with a (255, 223) Reed-Solomon outer code and either the TCM code or the rate 1/2 convolutional code as the inner code in the

presence of the KuSA RFI environment. This figure also shows the performance of the (255, 223) Reed-Solomon outer code by itself. (The inner code is an uncoded QPSK signal.) A comparison of Figure 8 with Figure 6 shows that a 10^{-5} BER can be achieved with minimal degradation with all of the concatenated codes and with the (255, 223) Reed-Solomon code by itself (no inner code).

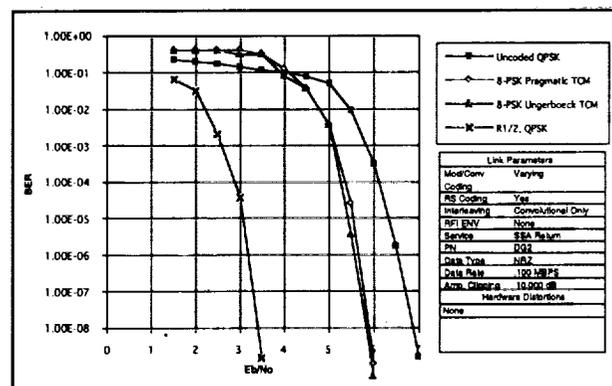


Figure 8. Performance of the Reed-Solomon Outer Code with Various Inner Codes in KuSA Return Service RFI

In the presence of Self-Interference. Figure 9 shows the TDRSS return service performance with the concatenated codes in the presence of an interfering signal, where the interfering signal has a lower symbol rate than the desired signal. It was assumed that the desired signal has a 7 dB E_b/N_0 margin, which is sufficient to ensure that noise is insignificant relative to the interference at high signal-to-interference ratios. This figure shows that the BER performance with a concatenated code and either the Ungerboeck or pragmatic TCM codes as the inner code is worse than the performance without an inner code. The Ungerboeck and the pragmatic TCM codes are more susceptible to decoding errors than an uncoded QPSK signal since the decision regions of an 8-PSK signal are closer together than the decision regions of an uncoded QPSK signal and the Reed-Solomon outer code cannot correct the decoding errors.

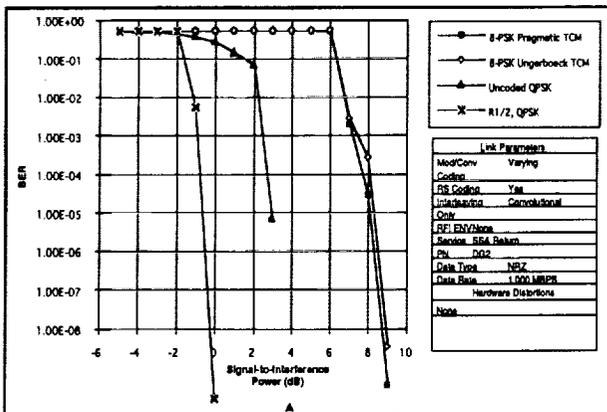


Figure 9. Performance of the (255, 223) Reed-Solomon Code with Different Inner Codes in the Presence of an Interfering Signal with a Lower Symbol Rate than the Desired signal

IV. CONCLUSIONS

The TCM codes without the (255, 223) Reed-Solomon outer code do not perform well in the presence of RFI or self-interference. The concatenated code which has an outer (255, 223) Reed-Solomon code and an inner TCM code can achieve a 10^{-5} BER in RFI, but its performance is worse than an uncoded QPSK signal in the presence of self-interference. Since RFI and self-interference are often present at S-Band frequencies, the TCM codes with or without the Reed-Solomon code outer code are not recommended for TDRSS S-Band return services. However, RFI and self-interference are not present as often on the Ku-Band return services as they are on the S-Band return services so that TCM codes with or without the Reed-Solomon outer code can be used for TDRSS Ku-Band return services.

V. REFERENCES

- [1] Ungerboeck, G., "Trellis-Coded Modulation with Redundant Signal Sets, Parts I and II", IEEE Communications Magazine, Vol 25, February 1987
- [2] Viterbi, Andrew J., Wolf, Jack K., Zehavi, Ephraim, Padovani, Roberto, "A Pragmatic Approach to Trellis-Coded Modulation", IEEE Communications Magazine, July 1989
- [3] Advanced Coding Techniques Study for TDRS/TDRS II Systems, Kaplan, Ted, Berman, Ted, STel 003R1493, April 23, 1993

NETWORK COMMAND PROCESSING SYSTEM OVERVIEW**Yonwoo Nam**

Senior Engineer

AlliedSignal Technical Services, Corp.
Columbia, Maryland 21045**Lisa D. Murphy**

Senior Scientific Systems Programmer

AlliedSignal Technical Services, Corp.
Columbia, Maryland 21045**ABSTRACT**

The Network Command Processing System (NCPS) developed for the National Aeronautics and Space Administration (NASA) Ground Network (GN) stations by the AlliedSignal Technical Services, Corporation (ATSC) is a spacecraft command system utilizing a MULTIBUS I/68030 microprocessor. This system was developed and implemented at ground stations worldwide to provide a Project Operations Control Center (POCC) with command capability for support of spacecraft operations such as the LANDSAT, Shuttle, Tracking and Data Relay Satellite and Nimbus-7. The NCPS consolidates multiple modulation schemes for supporting various manned/unmanned orbital platforms. The NCPS interacts with the POCC and a local operator to process configuration requests, generate modulated uplink sequences, and inform users of the ground command link status. This paper presents the system functional description, hardware description and the software design.

BACKGROUND

In 1987, ATSC was directed by the NASA to design a replacement spacecraft command system for the early 1970's vintage Spacecraft Command Encoder (SCE). The project goal was to devise a system that would utilize state-of-the-art hardware and software, provide on-site fault isolation and eliminate cumbersome analog alignment requirements. The initial design study determined that there would be a significant software cost benefit in modeling the system after the NASA Telemetry and Communications Data System (TCDS). The TCDS is a telemetry data processing system developed in the mid 1980s which utilized distributed processing techniques and was the replacement for outdated decommutators and data system computers on the NASA GN. Due to the differences in functionality between the TCDS and NCPS, only portions of the hardware components could be re-used. However, the basic hardware system architecture and the device driver software for these components could be implemented with minimal modification.

SYSTEM FUNCTIONAL DESCRIPTION

The NCPS provides the POCC and/or a local operator with the capability to command a spacecraft and to monitor the command system status. Figure 1 is a simplified functional block diagram of the system. There are three command modes: throughput, local, and hardline. Each mode has a different source of data. In throughput mode, data is received for uplink via the NASA Communication (NASCOM) Data Link in 4800 bit blocks. In local mode, data prestored in command pools is used. Each command pool may contain several spacecraft commands that are between 16 to 2000 bits in length. In hardline mode, serial command data is received via a connector on the rear of the NCPS and sent directly to the uplinking hardware with little intervention from the software.

The NCPS operator uses a menu interface to select one of the three commanding modes. Once a commanding mode has been selected, the operator enters a spacecraft identification (SID) code and a file designator character code. These two codes are used to retrieve the selected spacecraft attribute file from the hard disk which contains default commanding parameters. The NCPS loads this file into its database memory and enters a prepass state. The current configuration attributes, status information and menu selection items for modifying spacecraft attributes, is shown on the video terminal. Each commanding mode has two operating states: prepass and pass. Prepass is the state of operation prior to uplink. In this state, the operator can modify and override certain spacecraft attributes. Pass is the state of operation where data is uplinked to the spacecraft and is entered via an operator menu selection when the system is in the prepass state. In the pass state, some attributes that do not affect command uplink can be modified, including selecting and uplinking an idle data sequence between commands.

In the throughput commanding mode, three options are available: uplink immediate, buffered uplink, and rate adjust. The uplink immediate option permits the command data to be transmitted as soon as it is received. The buffered uplink option accumulates one command data block before beginning command transmission to the spacecraft. Buffering allows for compensation of irregularities in the data arrival time and a precisely metered continuous data flow to be generated. When the received average data rate is different than the precise uplink command data rate, a circumstance may arise in which the buffered data may be consumed or an excessive amount may accumulate. To diminish this possibility, a rate adjustment feature has been incorporated in which the uplink command rate is varied to match the average command rate received from the commanding source. If the number of blocks begins to decline, the software retards the uplink rate. If the number of blocks drops to zero, the software notifies the operator that an underflow has occurred. The software will also advance the uplink rate when the number of buffered blocks grows towards a predetermined maximum. The software notifies the operator that an overflow has occurred if the number of buffered blocks exceeds the predetermined maximum.

In the local commanding mode, the operator selects commands to be uplinked from a prestored command file by specifying a command "mark" number. This command will be uplinked

according to the parameters in the selected spacecraft attribute file upon release by the NCPS local operator.

In the hardline commanding mode, the operator configures the NCPS for the selected spacecraft and enters the pass state. Any data received through the hardline connection is uplinked according to the parameters in the selected spacecraft attribute file. This mode is used at locations where serial command data can be received directly from a POCC command generator. This mode is primarily used for testing a spacecraft shortly before launch.

The output of the NCPS is a Phase Shift Keyed (PSK) modulated subcarrier or a modulated squarewave for the scientific spacecraft and the shuttle, respectively which is routed to the transmitting system exciter. The subcarrier frequency and the data rate used to modulate the subcarrier are determined by the selected spacecraft attribute file.

The NCPS system status is reported using Site Status Messages (SSM) and command echo NASCOM blocks. Options to enable or disable both the SSM and echo block functions are available from the operator menu.

SSM blocks are generated and transmitted to the project specified by the status destination code at the rate of one block per second in both the prepass and pass state. SSMs contain information about the status of the NCPS including information about the site, identification of the spacecraft being commanded and the status of the uplink process.

Echo blocks contain the data that was uplinked and are transmitted to the project specified by the echo destination code. There are two types of echo blocks: asynchronous and synchronous. Asynchronous blocks contain the image of the uplink data as received from the verification receiver which samples the uplink RF output. Synchronous blocks are the NASCOM blocks that were received from a project using the source and destination codes interchanged and transmitted back to the originating project.

HARDWARE DESCRIPTION

The NCPS is housed in a single 19-in. standard NASA equipment rack. The units mounted in the rack include a 140 Mbyte hard disk drive, a 5¹/₄-inch floppy disk drive, a streaming tape unit, a 14-in. color graphics terminal, a standard 101 keyboard and the NCPS chassis. The functional and ergonomic lay-out provides the operators with easy access to the system. Figure 2 illustrates the NCPS rack elevation layout.

Chassis

The NCPS chassis uses a MULTIBUS I architecture with a 20-slot card cage to accommodate the six PC cards and to provide for expansion or modification. Of the six board assemblies, one is a commercially available computer board and five are special purpose custom designed boards

i.e. Serial Time Code Receiver board, Transmit/Receive board, External 5 MHz board, a PSK Modulation Board (PMB), and Shuttle Command Module (SCM).

Computer Board

The computer board utilizes a 68030 32-bit microprocessor with 4 Mbytes of RAM. This board performs the central control function of the NCPS. The system initialization instructions are stored in EPROM. The computer board also provides interfaces to disks, streaming tape, graphic video terminal and an external printer.

Serial Time Code Receiver Board

The Serial Time Code Receiver board was developed by NASA. The board decodes the received serial binary 1 (SB-1) time code which is a Manchester encoded RS-422 signal into parallel time with millisecond accuracy which the NCPS software inserts into the transmitted NASCOM 4800 bit block. The source of the SB-1 time code is the station master timing system.

Transmit/Receive Board

The Transmit/Receive (T/R) board was developed by ATSC for the TCDS and is used to interface with the NASCOM communication system. It provides the channel for processing a digital serial data stream of NASCOM blocks entering or leaving the NCPS.

External 5 MHz Board

The External 5 MHz board provides frequency synthesis by using a Phase-Locked Loop (PLL) technique. The purpose of the board is to generate a phase coherent 4.096 MHz signal from the station's 5 MHz signal. A monolithic PLL was used for fractional frequency synthesis in the External 5MHz board and consists of a Voltage Control Oscillator (VCO), phase comparator and low pass filter. The monolithic PLL was used in this application because of its low cost and high performance at frequencies below 50 MHz.

A block diagram representation of the fractional frequency synthesizer is shown in Figure 3. The phase locked loop operates by producing an oscillator frequency to match the frequency of an input signal. In this locked condition any slight change in input frequency, f_a , first appears as a change in phase between f_a and the oscillator frequency, f_c . The phase shift then acts as an error signal to change the oscillator's frequency to match the f_a .

Having a crystal-controlled VCO and phase-locked to the station's precise main timing system results in a long term stable clock. This procedure was incorporated in the hardware design to increase the stability of the modulated subcarrier.

PSK Modulation Board

The PSK Modulation Board (PMB) is designed to provide command support for all subcarrier modulated compatible spacecraft. Control, setup, and ground command verification sequences are received via the Multibus interface. The command data to be uplinked is received from the CPU via the Multibus interface in all modes with the exception that in hardline mode it is received via a direct serial interface.

The PMB is divided into three functional areas: Command Data Control (CDC), Subcarrier Modulation/Demodulation (SMD), and the Serial Data Interface (SDI). Figure 4 illustrates the PMB Functional Block Diagram.

The CDC interface permits control and setup of the PMB by the CPU via the multibus interface. The PMB setup/control words, provided from the hard disk's spacecraft attribute files, select the subcarrier frequency, data (modulation) rate, data type encoding, command idle, modulating source, and command mode.

The SMD process generates a composite modulated PSK waveform and utilizes a stable frequency source, rate multipliers, a sinusoidal look-up table, a digital-to-analog converter, and a single pole low-pass filter. The subcarrier rate multiplier along with the frequency reference, which can be from an on-board crystal or the External 5 MHz board, generates subcarrier samples at 256 times the selected subcarrier frequency. The subcarrier samples are the result of a phase counter addressing a sinusoidal look-up table, that is contained in PROM. The PROM contents are specified by the equation

$$D_i * \sin(2 * \pi * K/256)$$

where K represents the address of the subcarrier phase counter and D_i represents the sign of the data sequence. The active single pole low-pass filter eliminates the out of band harmonic power.

The SDI interface is provided to permit the processing of a serial command sequence. When the serial data mode is selected on the PMB, the transition tracking loop is selected versus the reference 4.096 MHz. The transition tracking loop drives the subcarrier phase to provide proper alignment of the subcarrier transitions and the data symbols being transmitted.

Shuttle Command Module

The Shuttle Command Module (SCM) is designed to provide the NCPS with a shuttle orbiter ground-to-space command link. A series of control data directives are used by the NCPS host processor to communicate to the SCM. The control data directives include uplink configuration, modulation source and rate.

The SCM can be configured to operate as both a voice-command multiplexer or as a throughput device. Figure 5 illustrates the SCM Functional Block Diagram. In a multiplexer configuration,

the SCM generates a multiplexed uplink sequence containing command patterns supplied by the selected source and overlays voice supplied from a local Delta Modulation Sub-system (DMS). The data sources in a multiplexer configuration are host, hardline, and local controller. As a throughput device, the selected source supplies the entire composite baseband uplink modulating sequence. The only valid data sources in the throughput configuration are host and hardline. In either configuration, multiplexer, or throughput, the composite baseband can be encoded with a rate 1/3 convolutional code with the following polynomial definitions:

$$G_1=D^6+D^3+D^2+D+1 ; G_2=D^6+D^5+D^3+D^2+D+1 ; G_3=D^6+D^4+D+1.$$

In support of hardline, analog tape playback, and host throughput rate adjust uplink, the SCM employs a digital tracking loop with a maximum tracking bandwidth of 200 ppm.

Peripheral

An external serial line printer provides a hard copy of all NCPS activities and status information and is used primarily for historical data and as a troubleshooting aid.

SOFTWARE DESCRIPTION

The NCPS software development task began in 1987. The original software project included software to support a wide range of ground stations and spacecraft. However, several ground stations were closing and some spacecraft were becoming obsolete. The NCPS was required to support both the aging spacecraft, as well as, future spacecraft. A design approach to maintain an ever-changing system was needed. Other projects were also faced with application software that was developed and modified on a continuing basis. In order to optimize the generation and maintenance of those applications a distributed operating system and a multi-tasking executive (MX) were developed to support these projects. A paper "Distributed Operating System for NASA Ground Station" written by John Doyle in 1987 provides background for the software section of this paper[1].

NCPS Software Design Goals and Considerations

The NCPS software design goal was to utilize as much of the in-house software and tools as possible without inhibiting the development and uniqueness of the NCPS application. Software design objectives described in the following paragraphs were considered during the design phase.

1. A modular approach in software development, allowing for incremental source code revision, was needed to reassure the growth of the NCPS. In order to support future spacecraft, such as the Space Shuttle, hardware devices and software drivers would need to be added to the NCPS baseline without disrupting a current working system.

2. The design was based on reusable and replaceable modules. As tools and software components evolved they could be substituted for older and less efficient ones. Modules that were used in other systems, such as, the Transmit/Receive board and driver software could be incorporated into the NCPS without modification.
3. A memory resident database containing information that could be accessed instantly was needed for updating status displays and the operator interface.
4. Software and tools developed during the NCPS design phase were written in C using the UNIX operating system on a Heurikon computer. In order to reuse software, a similar development environment was selected for the NCPS.
5. An objective to automate system documentation through the use of graphical representation was considered. An in-house software tool known as the Network Adaptive Schema for Modeling Asynchronous Computation (NASMAC) was developed for previous projects. This tool allowed for the specification of software systems using directed graphs, and the automatic transformation of such graphs into operational software. These graphs provided an overview of the application software without the knowledge of the underlying system. This tool was the foundation for the NCPS software. It provided application documentation in the form of directed graphs and a modular design for the software.
6. At the time of the NCPS design, an operator interface was developed in-house. Batch files, prompts, sequenced commands and a command processor for the menus, along with the database, display formats and a display processor, facilitated an operator interface that could be custom designed on the fly. No recompilation of code was necessary. The system database could be instantly accessed and updated, providing up-to-date status information.

The objectives that were formed proved to be advantageous in the growth of the NCPS and the support of future spacecraft. Operator interface software, in-house developed tools and device driver software could be reused while new components and modules could be added.

NCPS Modular Software Design

Once the decision to use a modular design approach and in-house developed software was made, focus was moved to the NCPS functionality and the software application. Figure 6 is a data flow diagram which depicts the NCPS application and the software components. Each node on the data flow diagram depicts a module in the NCPS application. The following is a description of each of the modules.

1. Operator Interface: This module allows for a local operator to communicate with the NCPS system. Drivers to support terminal and printer devices, alarm functions which provide the operator with information about the state of the system, and the operator interface command and

display processors were reused from previous projects. New command files and ASCII display formats were created to support the NCPS specific menus.

2. Local Commanding and Utilities: Local Commanding software to support this module was written specifically for the NCPS. It included a means to create and update command pools, test files and attribute files. Attribute files are used to configure the NCPS in order to support a variety of stations and spacecraft. Run-time hard disk file system utilities, developed for previous projects, were incorporated into the NCPS.

3. NASCOM Interface: The NASCOM interface consists of a driver to support the Transmit/Receive board that has been used in several projects. The code and hardware for this module was incorporated in the NCPS without modification. This board receives serial data from the NASCOM Data Link (NDL) and blocks it in the form of NASCOM blocks. It also extracts data from a NASCOM block and serially transfers it to the NDL.

4. Throughput Commanding: Throughput commanding for the NCPS software performs verification of the blocks received from the NASCOM interface and passes it to the PSK Modulator/Demodulator (MOD/DEMOM) interface for uplink. All code for this module was written specifically for the NCPS.

5. PSK MOD/DEMOM Interface: The PSK MOD/DEMOM interface is device driver to support the PSK Modulation Board. It transfers forward and configuration data to the PSK board and receives echo and status information from the PSK board.

6. ECHO/STATUS Process: This module collects echo and status data and formats it into a NASCOM block. It creates a header based on information stored in the spacecraft attribute file.

NCPS Space Shuttle Modification

Because of the modular design of the system, the NCPS was able to incorporate software to support the Space Shuttle in approximately 6 person-months without disrupting the current working system. Modification to the NCPS software included new command files and display formats to support a Shuttle specific operator interface. Routines to support forward, echo and status messages specifically for the Space Shuttle were added. New attribute files were created to support system configuration for Johnson Space Center (JSC), Tape, and Emergency Voice Command Fill (EVCF) commanding. The "PSK MOD/DEMOM Interface" in figure 6 was replaced with the "SCM MOD/DEMOM Interface" which included new device driver software for the Shuttle Command Module (SCM).

SUMMARY

After detailed research and analysis, the NCPS was developed and implemented to provide GN sites with command capability for support of spacecraft operations. The modularization and

commonality of parts have help produced a system that can be expanded as needed. Software and hardware modules can be added to the NCPS as requirements to support future sites and spacecraft are identified.

ACKNOWLEDGEMENT

The hardware and software, with the exception of NASMAC described herein were developed on NASA contract NAS5-31000 for the Goddard Space Flight Center's Networks Division.

REFERENCES

- [1] "Distributed Operating System for NASA Ground Stations", Doyle, John F., International Telemetry Conference 1987, San Diego, CA Volume XXIII, p425.

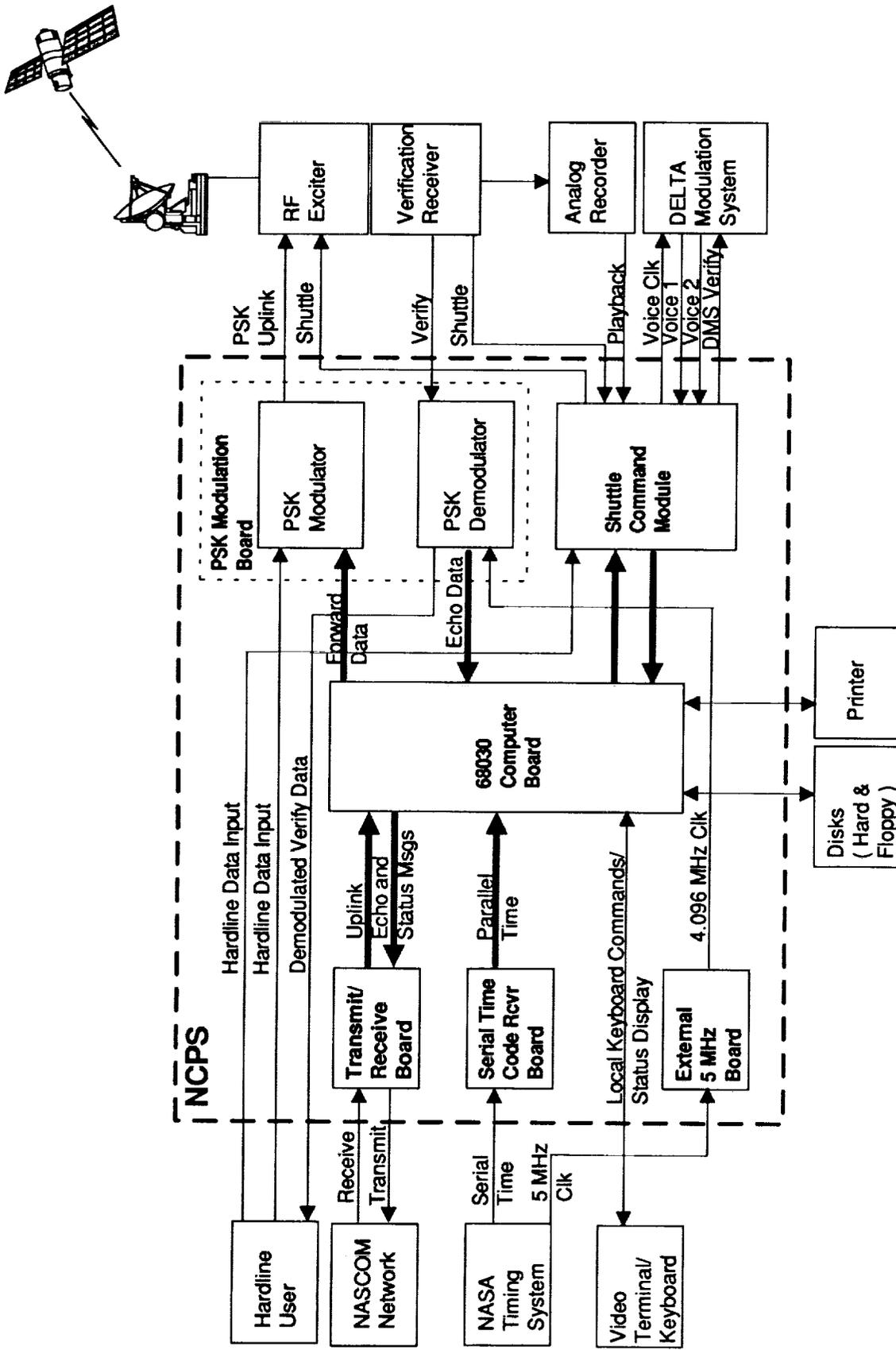


FIGURE 1. NCPS FUNCTIONAL BLOCK DIAGRAM

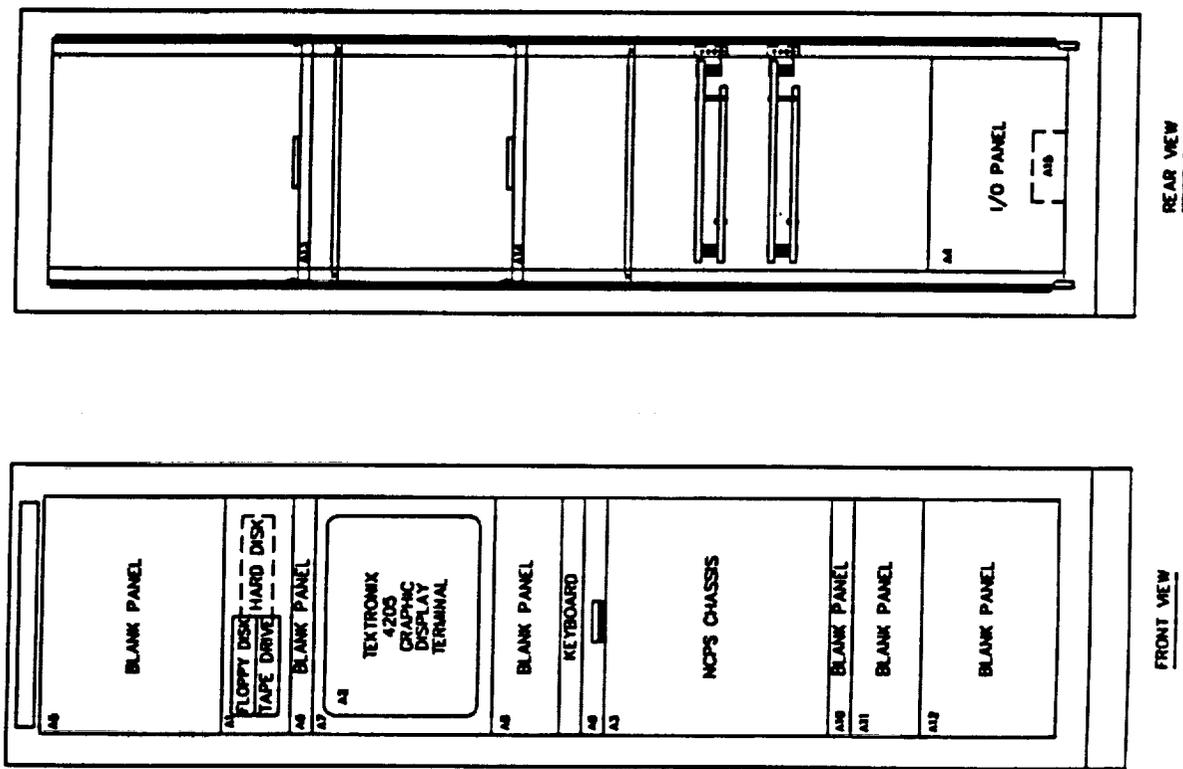
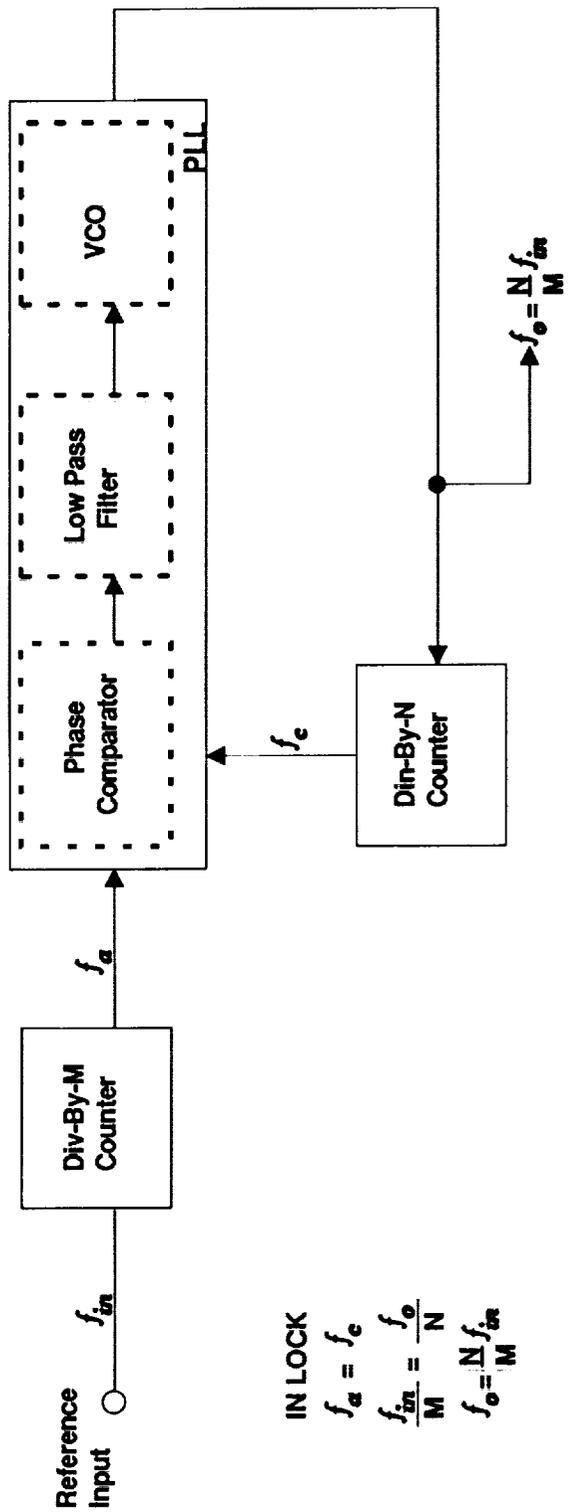


FIGURE 2. NCPS RACK ELEVATION



IN LOCK

$$f_a = f_c$$

$$\frac{f_{in}}{M} = \frac{f_o}{N}$$

$$f_o = \frac{N}{M} f_{in}$$

FIGURE 3. FRACTIONAL FREQUENCY SYNTHESIZER

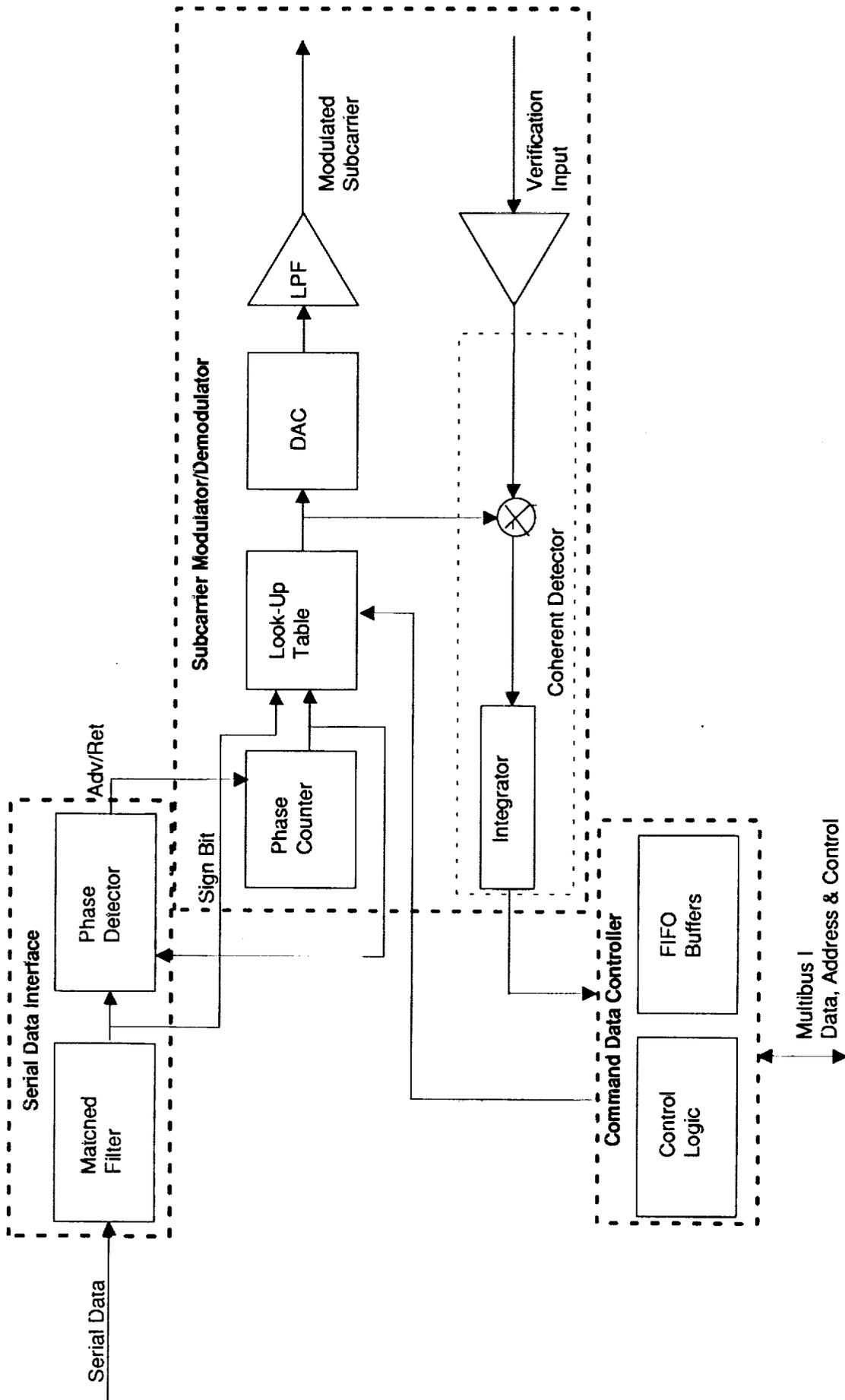


FIGURE 4. PMB FUNCTIONAL BLOCK DIAGRAM

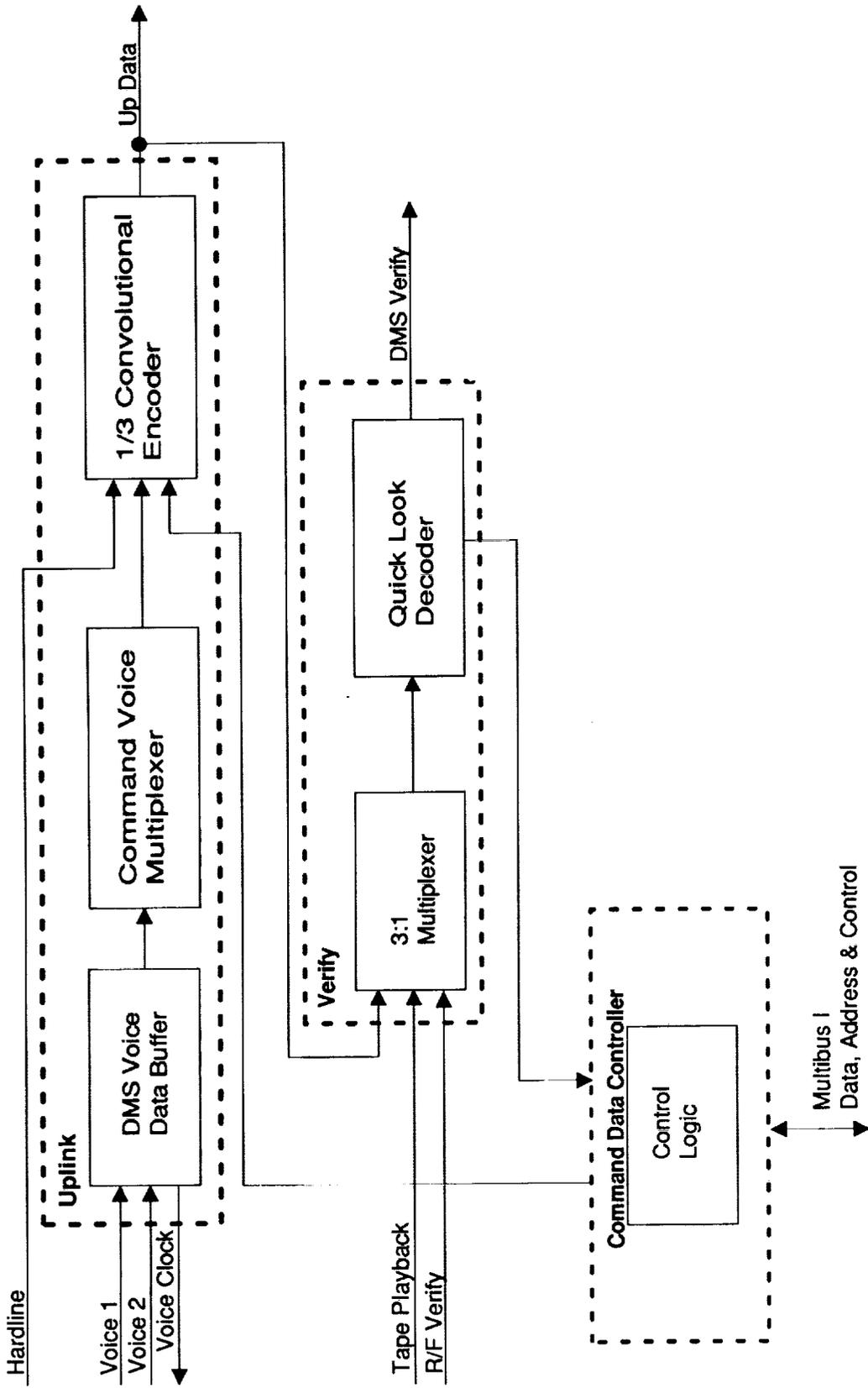


FIGURE 5. SCM FUNCTIONAL BLOCK DIAGRAM

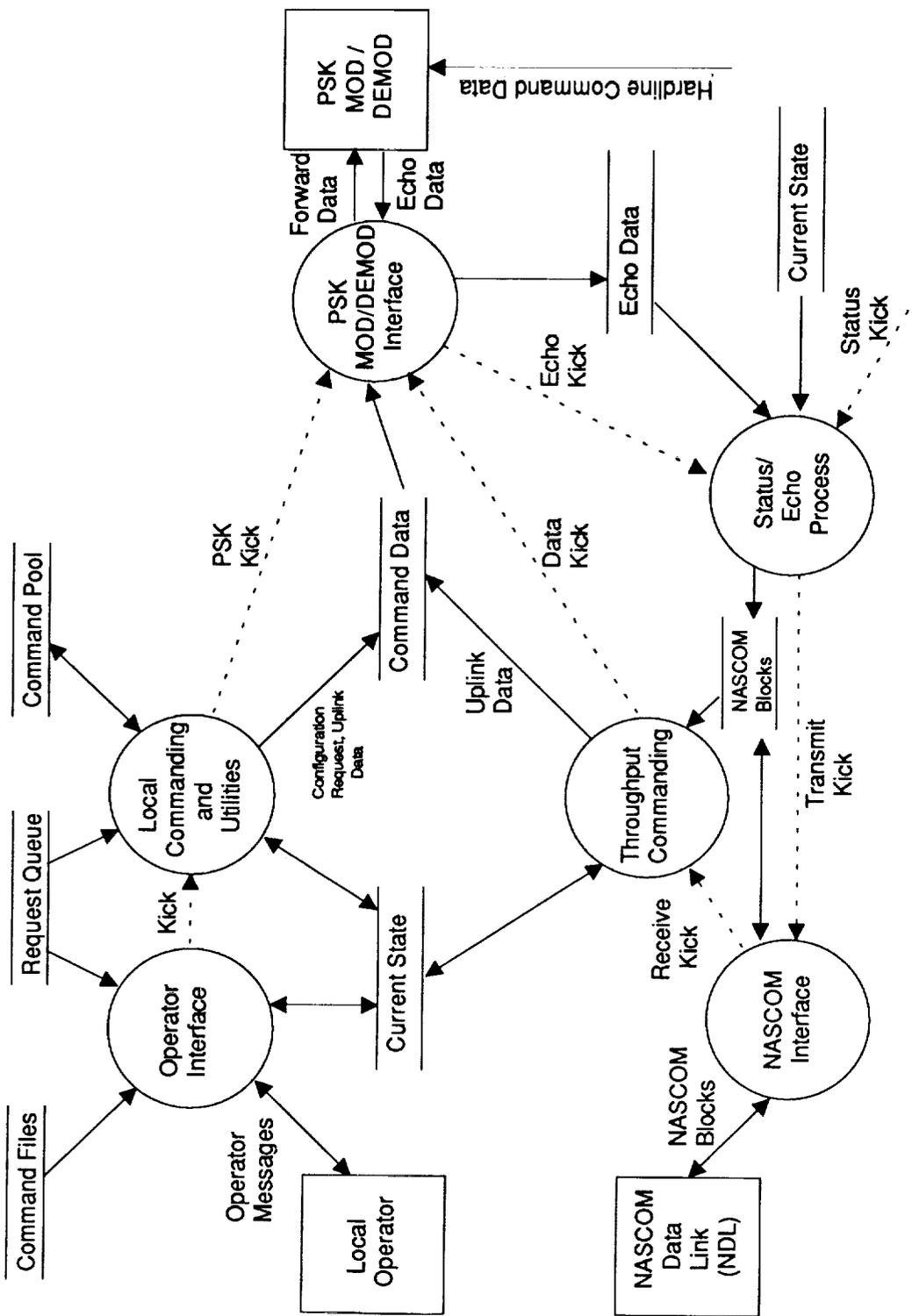


FIGURE 6. NCPS DATA FLOW DIAGRAM

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE November 1993	3. REPORT TYPE AND DATES COVERED Conference Publication	
4. TITLE AND SUBTITLE Networks Technology Conference		5. FUNDING NUMBERS Code 535	
6. AUTHOR(S) Edited by Keiji Tasaki			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Goddard Space Flight Center Greenbelt, Maryland 20771		8. PERFORMING ORGANIZATION REPORT NUMBER 94B00011	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CP-3241	
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 62 Report available from the NASA Center for AeroSpace Information, 800 Elkridge Landing Road, Linthicum Heights, MD 21090; (301) 621-0390.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The papers included in these proceedings represent the most interesting and current topics being pursued by personnel of the GSFC Networks Division and supporting contractors involved in Space, Ground, and Deep Space Network (DSN) technical work. Although 29 papers are represented in the proceedings, only 12 were presented at the conference, because of space and time limitations. Proceedings are organized according to five principal technical areas of interest to the Networks Division: Project Management; Network Operations; Network Control, Scheduling, and Monitoring; Modeling and Simulation; and Telecommunications Engineering.			
14. SUBJECT TERMS Project Management, Network Operations, Network Control, Scheduling and Monitoring, Modeling and Simulation, Telecommunications Engineering		15. NUMBER OF PAGES 374	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited